



ESCUELA SUPERIOR DE INGENIERÍA

*INGENIERÍA TÉCNICA EN INFORMÁTICA DE  
GESTIÓN*

APLICACIÓN PARA ENTRENADORES DE FÚTBOL SALA  
(TRAININGFS)

Rafael Gamero del Río

11 de Abril de 2014





ESCUELA SUPERIOR DE INGENIERÍA DE CÁDIZ

INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

APLICACIÓN PARA ENTRENADORES DE FÚTBOL SALA  
(TRAININGFS)

- Departamento: Ingeniería Informática
- Director del proyecto: Iván Ruiz Rube
- Alumno: Rafael Gamero del Río

Cádiz, 11 de Abril de 2014

FDO: Rafael Gamero del Río



# Agradecimientos

A la finalización de este Proyecto Fin de Carrera, que concluye una etapa de mi vida, quiero expresar un profundo agradecimiento a cada persona que me encontré en el camino para llegar hasta aquí, ya que han sido parte vital de mi crecimiento personal y profesional.

Agradecer encarecidamente a Sergio Barroso por concebir la idea de software, pues sin ella no podría haberse hecho realidad este proyecto y por ofrecerme una visión diferente de este deporte. Alabar además su paciente escucha a cada detalle que le he propuesto.

Agradecer a Iván Ruiz Rube, coordinador del proyecto, su labor guiando mi trabajo y aprendizaje. Éstos meses con él me han hecho sentir capacitado para afrontar por mí mismo las dificultades de un proyecto real.

Agradecer a mi familia su apoyo constante, motivación e incluso insistencia para trabajar cada día en este proyecto.

Sin más, agradecer a todas las personas que han estado a mi lado durante todos estos años. Por ellos soy quien soy.

GRACIAS



# Resumen

A modo de sugerencia surgió la idea del software *TrainingFS*, una aplicación web para entrenadores de fútbol sala, propuesta de un técnico allegado. Diversos factores han provocado que un gran sector de este colectivo no haya encontrado aun en la tecnología un soporte que les facilite su labor. Destacan entre estas carencias la complejidad de los sistemas existentes que requerirían el trabajo conjunto de un cuerpo técnico propio de los grandes equipos o los elevados costes para obtener la licencia de uso. Ambas razones han provocado que la utilización de programas informáticos para entrenadores estén al alcance de muy pocos.

Basándonos en estas necesidades, el usuario medio del gremio y las tendencias tecnológicas actuales, se propuso la idea de enfocar el proyecto hacia una aplicación web que, sin apenas restricciones, un entrenador de cualquier categoría pueda utilizar. Por ello, se decidió usar un framework de PHP, basado en la filosofía Modelo-Vista-Controlador, para desarrollar y alojar en la red un software sencillo y accesible que permitiera planificar, mantener e imprimir sesiones de entrenamiento, acompañado de un diseñador de tareas semejante a las pizarras que están acostumbrados a utilizar, emulada con la novedosa etiqueta “canvas” de HTML5.

Varias fueron las reuniones necesarias para definir los requisitos del sistema, acercándonos un poco más en cada una de ellas, de forma que el software cumpliera las necesidades de cualquier entrenador de este deporte. Al igual que la especificación, todo el proceso de desarrollo ha seguido un ciclo incremental. Marcando el final de cada iteración podemos situar una valoración del progreso realizado por parte del “cliente”. Durante las primeras fases esta valoración significaba una comunicación verbal que llevara al entendimiento de ambas partes y, posteriormente, suponía la utilización del incremento del sistema. Por su parte, todas estas iteraciones han comenzado con un periodo de aprendizaje correspondiente a las tareas a realizar en ella. Estos procesos didácticos han supuesto el conocimiento de distintas tecnologías, lenguajes y herramientas necesarias para llevar a cabo la ingeniería del software, siempre haciendo uso de software libre. De este modo, nos hemos asegurado que el coste del sistema no aumentara debido al uso de herramientas comerciales y ha abierto un amplio abanico de posibilidades para proyectos futuros. Aunque en contraposición a estos beneficios, cabe destacar que los numerosos procesos de aprendizaje han provocado un retraso significativo en la planificación del proyecto.

Una vez implementado el software y como parte del proceso de pruebas, se le ha facilitado la aplicación a una serie de entrenadores para obtener la opinión de usuarios potenciales. Este hecho nos ha facilitado información de gran utilidad, ya que nos han servido de algo más que simples testadores. La opinión recogida en este proceso ha permitido confeccionar un manual de usuario gráfico, intuitivo y comprensible, haciendo hincapié en los aspectos más problemáticos de uso del sistema.

### **Palabras clave**

Aplicación web  
Fútbol sala  
Entrenador  
Entrenamiento  
Pizarra  
Campograma



# Índice general

<b>I</b>	<b>Prolegómeno</b>	<b>14</b>
<b>1.</b>	<b>Introducción</b>	<b>16</b>
1.1.	Motivación . . . . .	16
1.2.	Objetivos y alcance del proyecto . . . . .	16
1.3.	Organización del documento . . . . .	17
<b>2.</b>	<b>Planificación</b>	<b>18</b>
2.1.	Metodología de desarrollo . . . . .	18
2.2.	Planificación del proyecto . . . . .	19
2.3.	Organización . . . . .	23
2.4.	Costes . . . . .	23
2.5.	Gestión de riesgos . . . . .	24
<b>II</b>	<b>Desarrollo</b>	<b>26</b>
<b>3.</b>	<b>Análisis de requisitos</b>	<b>28</b>
3.1.	Catálogo de actores . . . . .	28
3.2.	Requisitos funcionales . . . . .	28
3.3.	Requisitos de información . . . . .	56
3.4.	Requisitos no funcionales . . . . .	58
3.5.	Reglas de negocio . . . . .	58
3.6.	Estudio de las alternativas tecnológicas . . . . .	59
<b>4.</b>	<b>Diseño del sistema</b>	<b>64</b>
4.1.	Diseño de la arquitectura . . . . .	64
4.1.1.	Arquitectura física . . . . .	64
4.1.2.	Arquitectura lógica . . . . .	65
4.1.3.	Arquitectura de diseño . . . . .	67
4.2.	Diseño de la interfaz de usuario . . . . .	68
4.3.	Diseño de datos . . . . .	72
4.4.	Diseño de componentes . . . . .	80
<b>5.</b>	<b>Implementación del sistema</b>	<b>102</b>
5.1.	Entorno tecnológico . . . . .	102
5.2.	Código fuente . . . . .	103

<b>6. Pruebas del sistema</b>	<b>108</b>
6.1. Pruebas unitarias . . . . .	108
6.2. Pruebas de integración . . . . .	108
6.3. Pruebas del sistema . . . . .	109
6.3.1. Pruebas funcionales . . . . .	109
6.3.2. Pruebas no funcionales . . . . .	111
6.4. Pruebas de aceptación . . . . .	113
 <b>III Epílogo</b>	 <b>114</b>
<b>7. Manual de usuario</b>	<b>116</b>
7.1. Introducción . . . . .	116
7.2. Características . . . . .	116
7.3. Requisitos previos . . . . .	117
7.4. Utilización . . . . .	117
 <b>8. Manual de instalación</b>	 <b>125</b>
8.1. Introducción . . . . .	125
8.2. Requisitos previos . . . . .	125
8.3. Procedimientos de instalación . . . . .	126
8.4. Pruebas de implantación . . . . .	128
 <b>9. Conclusiones</b>	 <b>130</b>
9.1. Objetivos . . . . .	130
9.2. Lecciones aprendidas . . . . .	130
9.3. Trabajo futuro . . . . .	131

# Índice de figuras

2.1. Modelo de ciclo de vida . . . . .	19
2.2. Estimación temporal. Diagrama de Gantt . . . . .	20
2.3. Calendario operativo. Diagrama de Gantt . . . . .	21
3.1. Diagrama casos de uso: Sistema . . . . .	29
3.2. Diagrama casos de uso: Sistema (extendido) . . . . .	29
3.3. Diagrama casos de uso: Gestión de campograma . . . . .	30
3.4. Diagrama casos de uso: Gestión de información . . . . .	36
3.5. Diagrama casos de uso: Gestión de tareas . . . . .	37
3.6. Diagrama casos de uso: Gestión de estrategias . . . . .	42
3.7. Diagrama casos de uso: Gestión personal . . . . .	47
3.8. Diagrama casos de uso: Gestión de planificación . . . . .	52
3.9. Diagrama conceptual de clases . . . . .	57
4.1. Arquitectura física . . . . .	64
4.2. Arquitectura lógica . . . . .	65
4.3. Comparación patrón MVC - Layers . . . . .	67
4.4. Flujo de navegación . . . . .	68
4.5. Diseño de interfaz - Login . . . . .	69
4.6. Diseño de interfaz - Registro . . . . .	69
4.7. Diseño de interfaz - Home . . . . .	71
4.8. Diseño de datos . . . . .	73
4.9. Registrar usuario . . . . .	82
4.10. Identificar usuario . . . . .	83
4.11. Editar perfil de usuario . . . . .	84
4.12. Listar tareas . . . . .	86
4.13. Visualizar tarea . . . . .	87
4.14. Registrar tarea . . . . .	88
4.15. Eliminar tarea . . . . .	89
4.16. Listar objetivos . . . . .	91
4.17. Registrar objetivo . . . . .	92
4.18. Eliminar objetivo . . . . .	93
4.19. Visualizar sesión . . . . .	95
4.20. Guardar sesión . . . . .	96
4.21. Imprimir sesión . . . . .	97
4.22. Visualizar estadísticas . . . . .	98

4.23. Modificar cuentas . . . . .	99
4.24. Visualizar ayuda . . . . .	100
7.1. Identificación de usuario . . . . .	118
7.2. Registro de usuario . . . . .	118
7.3. Acceso al sistema . . . . .	119
7.4. Registrar objetivo . . . . .	120
7.5. Visualizar tarea . . . . .	121
7.6. Guardar tarea . . . . .	122
7.7. Registro sesión . . . . .	123
7.8. Visualizar sesión . . . . .	124

# Índice de tablas

2.1. Costes de personal estimado . . . . .	20
2.2. Costes de personal real . . . . .	22
3.1. Caso de uso: Crear Campograma . . . . .	30
3.2. Caso de uso: Seleccionar pista . . . . .	31
3.3. Caso de uso: Incluir elemento . . . . .	31
3.4. Caso de uso: Mover elemento . . . . .	32
3.5. Caso de uso: Eliminar elemento . . . . .	32
3.6. Caso de uso: Rotar elemento . . . . .	33
3.7. Caso de uso: Trazar línea . . . . .	33
3.8. Caso de uso: Borrar línea . . . . .	34
3.9. Caso de uso: Editar color línea . . . . .	34
3.10. Caso de uso: Incluir conector . . . . .	34
3.11. Caso de uso: Eliminar conector . . . . .	35
3.12. Caso de uso: Editar color conector . . . . .	35
3.13. Caso de uso: Registrar tarea . . . . .	38
3.14. Caso de uso: Listar tareas . . . . .	38
3.15. Caso de uso: Visualizar tarea . . . . .	39
3.16. Caso de uso: Eliminar tarea . . . . .	39
3.17. Caso de uso: Editar tarea . . . . .	39
3.18. Caso de uso: Registrar objetivo . . . . .	40
3.19. Caso de uso: Editar objetivo . . . . .	40
3.20. Caso de uso: Listar objetivos . . . . .	41
3.21. Caso de uso: Eliminar objetivo . . . . .	41
3.22. Caso de uso: Registrar estrategia . . . . .	43
3.23. Caso de uso: Listar estrategias . . . . .	43
3.24. Caso de uso: Visualizar estrategia . . . . .	44
3.25. Caso de uso: Eliminar estrategia . . . . .	44
3.26. Caso de uso: Editar estrategia . . . . .	44
3.27. Caso de uso: Registrar acción . . . . .	45
3.28. Caso de uso: Editar acción . . . . .	45
3.29. Caso de uso: Listar acciones . . . . .	46
3.30. Caso de uso: Eliminar acción . . . . .	46
3.31. Caso de uso: Registrar usuario . . . . .	48
3.32. Caso de uso: Identificar usuario . . . . .	48
3.33. Caso de uso: Eliminar usuario . . . . .	49

3.34. Caso de uso: Editar usuario . . . . .	49
3.35. Caso de uso: Registrar club . . . . .	50
3.36. Caso de uso: Listar clubes . . . . .	50
3.37. Caso de uso: Visualizar club . . . . .	50
3.38. Caso de uso: Eliminar club . . . . .	51
3.39. Caso de uso: Editar club . . . . .	51
3.40. Caso de uso: Registrar sesión . . . . .	52
3.41. Caso de uso: Listar sesiones . . . . .	53
3.42. Caso de uso: Visualizar sesión . . . . .	53
3.43. Caso de uso: Editar sesión . . . . .	54
3.44. Caso de uso: Eliminar sesión . . . . .	54
3.45. Caso de uso: Imprimir sesión . . . . .	55
6.1. Caso de prueba: Identificación de usuario correcta . . . . .	109
6.2. Caso de prueba: Identificación de usuario errónea . . . . .	109
6.3. Caso de prueba: Registro de tarea . . . . .	110
6.4. Caso de prueba: Registro de tarea sin objetivos . . . . .	110
6.5. Caso de prueba: Registro de tarea incorrecto . . . . .	110
6.6. Caso de prueba: Visualizar estadísticas . . . . .	110

Parte I

Prolegómeno





# Capítulo 1

## Introducción

### 1.1. Motivación

La principal motivación radica en satisfacer las necesidades de Sergio Barroso, entrenador del Cádiz CF - Virgili FS y seleccionador del equipo universitario de Fútbol Sala masculino de la propia Universidad de Cádiz, de los que formo parte como jugador. Por otro lado, debemos mencionar que existen en el mercado programas y/o aplicaciones que podrían resultar útil en este ámbito, pero no se trata de software libre.

### 1.2. Objetivos y alcance del proyecto

Como objetivo principal del sistema tenemos la planificación y posterior impresión de las sesiones de entrenamiento, de modo que sirva de guía al entrenador durante el desarrollo de estas. En este sentido, una sesión reflejará entre otros aspectos el lugar físico donde se llevará a cabo o el binomio fecha-hora en que se realizará y el conjunto de tareas que se pondrán en práctica durante el desarrollo de esta. Como objetivo secundario y, a su vez, antecesor al mencionado destacamos el diseño de tareas para su inclusión en dichas sesiones. El diseño de tareas supondrá describirla tanto gráficamente como escrita. Por otra parte, debemos nombrar un objetivo paralelo a estos, el diseño de estrategias que al igual que las tareas serán descritas gráfica y textualmente.

El alcance del proyecto es máximo, pues se espera el cumplimiento íntegro de los objetivos del proyecto. La forma en que se prevé cumplir con esto, será desarrollando un sistema que proporcione funcionalidades para los objetivos nombrados.

## 1.3. Organización del documento

La presente memoria consta del siguiente contenido:

### I PROLEGÓMENO

#### 1. INTRODUCCIÓN

Describe la motivación del presente proyecto, la descripción del sistema actual (si existiese), los objetivos y cómo se ha estructurado la documentación.

#### 2. PLANIFICACIÓN

En esta sección se describen todos los aspectos relativos a la planificación del proyecto: metodología, organización, costes, planificación y gestión de riesgos.

### II DESARROLLO

#### 3. ANÁLISIS DE REQUISITOS

Se detallan los actores del sistema, los requisitos funcionales, los requisitos de información, los requisitos no funcionales y las reglas de negocio. Luego se describen las diferentes alternativas tecnológicas.

#### 4. DISEÑO DEL SISTEMA

Recoge la arquitectura general del sistema de información, el diseño de la interfaz de usuario, el diseño físico de datos, el diseño de componentes software.

#### 5. IMPLEMENTACIÓN DEL SISTEMA

Trata sobre todos los aspectos relacionados con la implementación del sistema en código, haciendo uso de un determinado entorno tecnológico.

#### 6. PRUEBAS DEL SISTEMA

En este capítulo se documentan los diferentes tipos de pruebas que se han llevado a cabo, ya sean manuales (mediante listas de comprobación) o automatizadas mediante algún software específico de pruebas.

### III EPÍLOGO

#### 7. MANUAL DE USUARIO

Se detallan las instrucciones de uso del sistema.

#### 8. CONCLUSIONES

En este último capítulo se detallan las lecciones aprendidas tras el desarrollo del presente proyecto y se identifican las posibles oportunidades de mejora sobre el software desarrollado.

#### BIBLIOGRAFÍA

#### INFORMACIÓN SOBRE LICENCIA

#### ANEXO I - INFORME DE MODERN.IE

#### ANEXO II - REPORTE DE GTMETRIX

#### ANEXO III - INFORME DE USUARIOS

## Capítulo 2

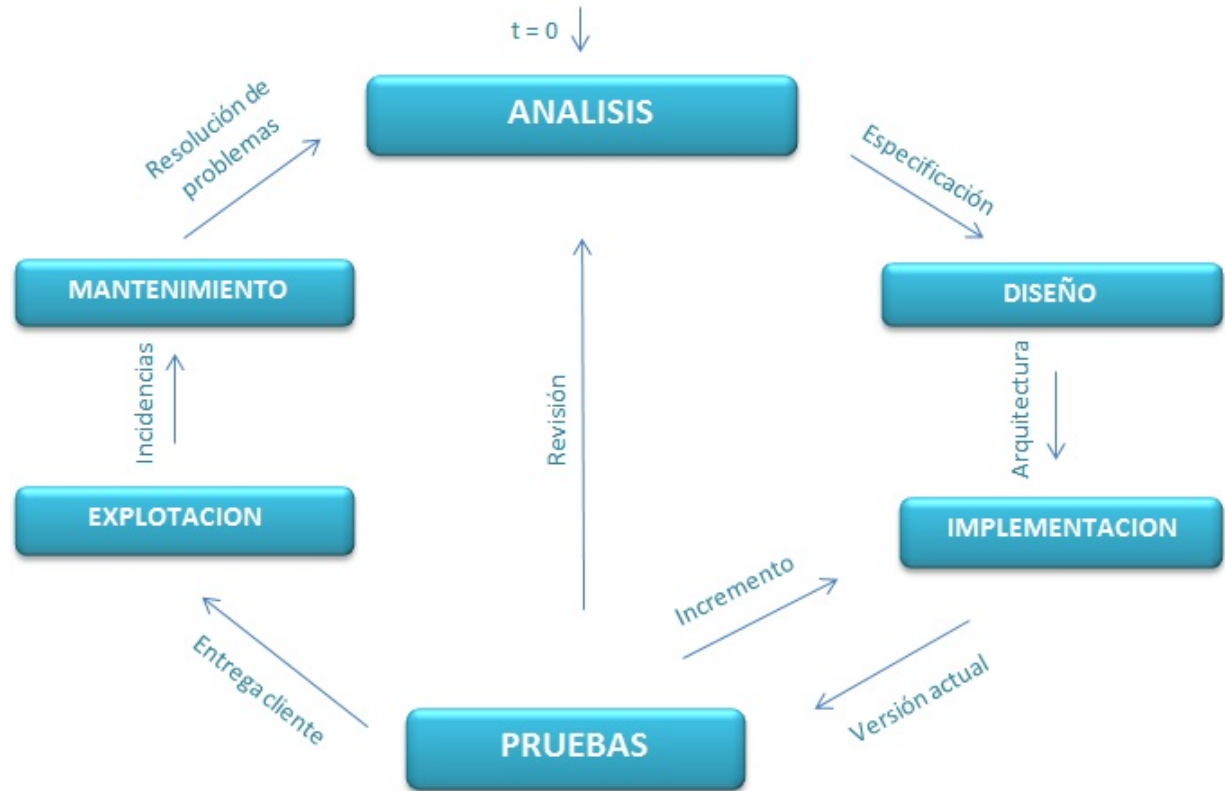
# Planificación

### 2.1. Metodología de desarrollo

El estándar ISO/IEC 12207-1 define el modelo de ciclo de vida como “*un marco de referencia que contiene los procesos, actividades y tareas involucradas en el desarrollo, explotación y mantenimiento de un producto software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso*”.

En nuestro caso, que podemos ver en la figura 2.1, empleamos una metodología de desarrollo estructurada y ágil, es decir, desarrollo iterativo e incremental. Por tanto, el ciclo de vida queda resumido en un conjunto de iteraciones. Las fases del ciclo de vida son “*Análisis*”, “*Diseño*”, “*Implementación*”, “*Pruebas*”, “*Explotación*” y “*Mantenimiento*”. Tras realizar el análisis (obtenemos la especificación de requisitos) y el diseño de la aplicación (definimos la arquitectura del sistema), comenzaremos la implementación o codificación del sistema, obteniendo una versión inicial muy limitada. Dicha versión será probada y en el caso de obtener resultados positivos, se pasará de nuevo a la fase de implementación donde se conseguirá una versión ampliada de la anterior. Este paso ha de repetirse tantas veces como sea necesario hasta obtener la versión final para que el cliente la utilice, pasando de este modo a la fase de explotación. Si, en su caso, las pruebas no dan los resultados esperados, habrá que volver a la fase de análisis para averiguar los problemas relativos a esta versión y reconducirla. Finalmente, durante la explotación del sistema pueden aparecer errores o mejoras necesarias para éste. Será entonces cuando pasemos a la fase de mantenimiento, donde se redactarán estas incidencias antes de pasar de nuevo a la fase de análisis. De este modo, podemos decir que nuestro modelo es incremental e iterativo.

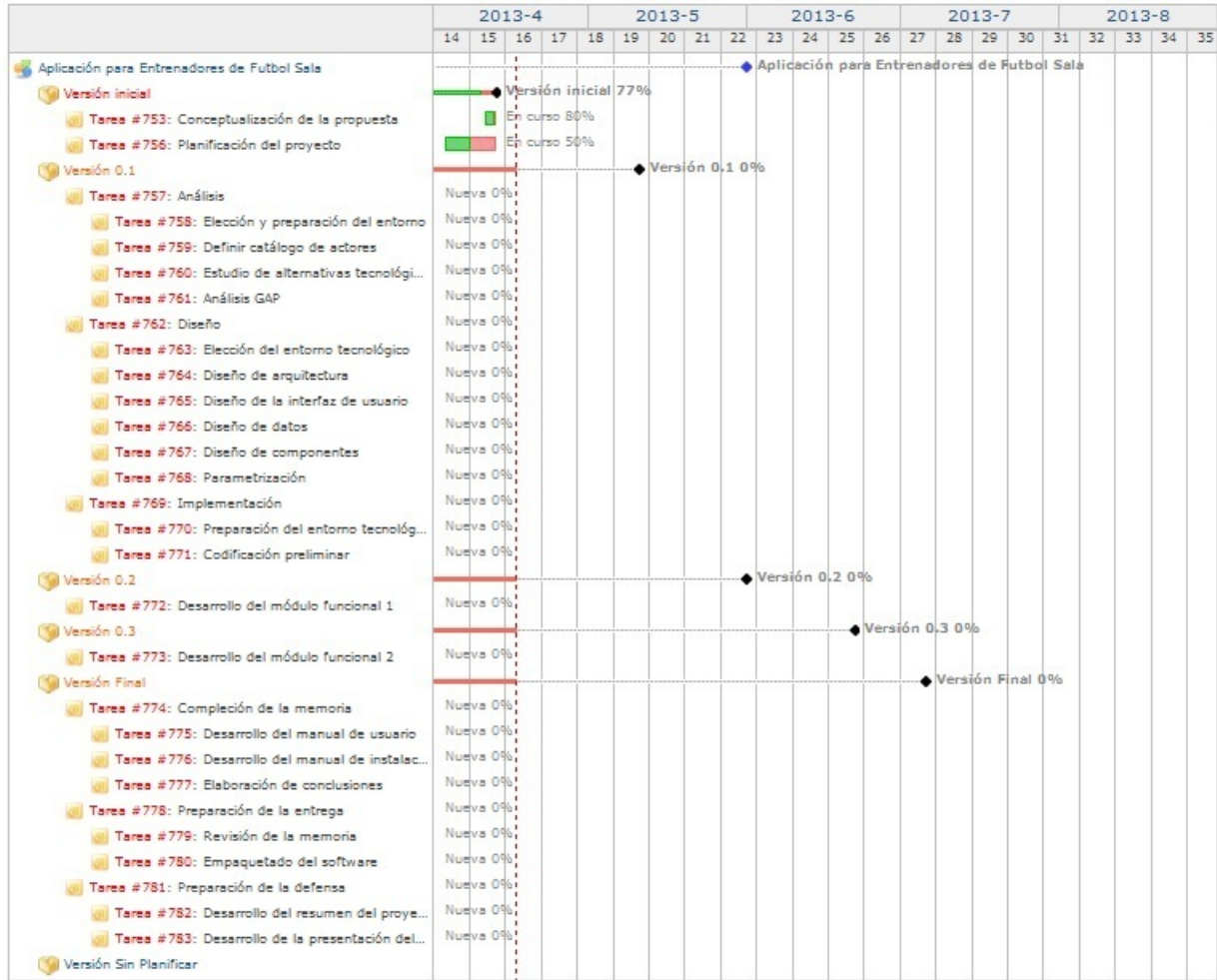
Figura 2.1: Modelo de ciclo de vida



## 2.2. Planificación del proyecto

La planificación temporal referidas a las actividades del proceso de desarrollo ha sido obtenida de la herramienta *redmine* que ofrece una guía para el desarrollo de proyectos fin de carrera de la Universidad de Cádiz. De modo que dicha estimación a día 16 de Abril de 2013 queda recogida en el diagrama de Gantt figura 2.2:

Figura 2.2: Estimación temporal. Diagrama de Gantt



Por su parte, la planificación temporal en relación al personal involucrado ha sido calculado a partir de estimaciones en número de horas que el desarrollador va a dedicar al proyecto. Podemos ver un desglose del volumen de horas en la tabla 2.1. Se debe mencionar que se ha considerado que la semana laboral por convenio está compuesta de 40 horas.

Tabla 2.1: Costes de personal estimado

Mes	Horas/Semana	Semanas	Personas/Mes (P/M)
Abril	25	4	$\frac{25}{40} * pers/mes = 0,625 * pers/mes$
Mayo	30	4	$\frac{30}{40} * pers/mes = 0,75 * pers/mes$
Junio	30	4	$\frac{30}{40} * pers/mes = 0,75 * pers/mes$
Julio	30	4	$\frac{30}{40} * pers/mes = 0,75 * pers/mes$

Horas estimadas:

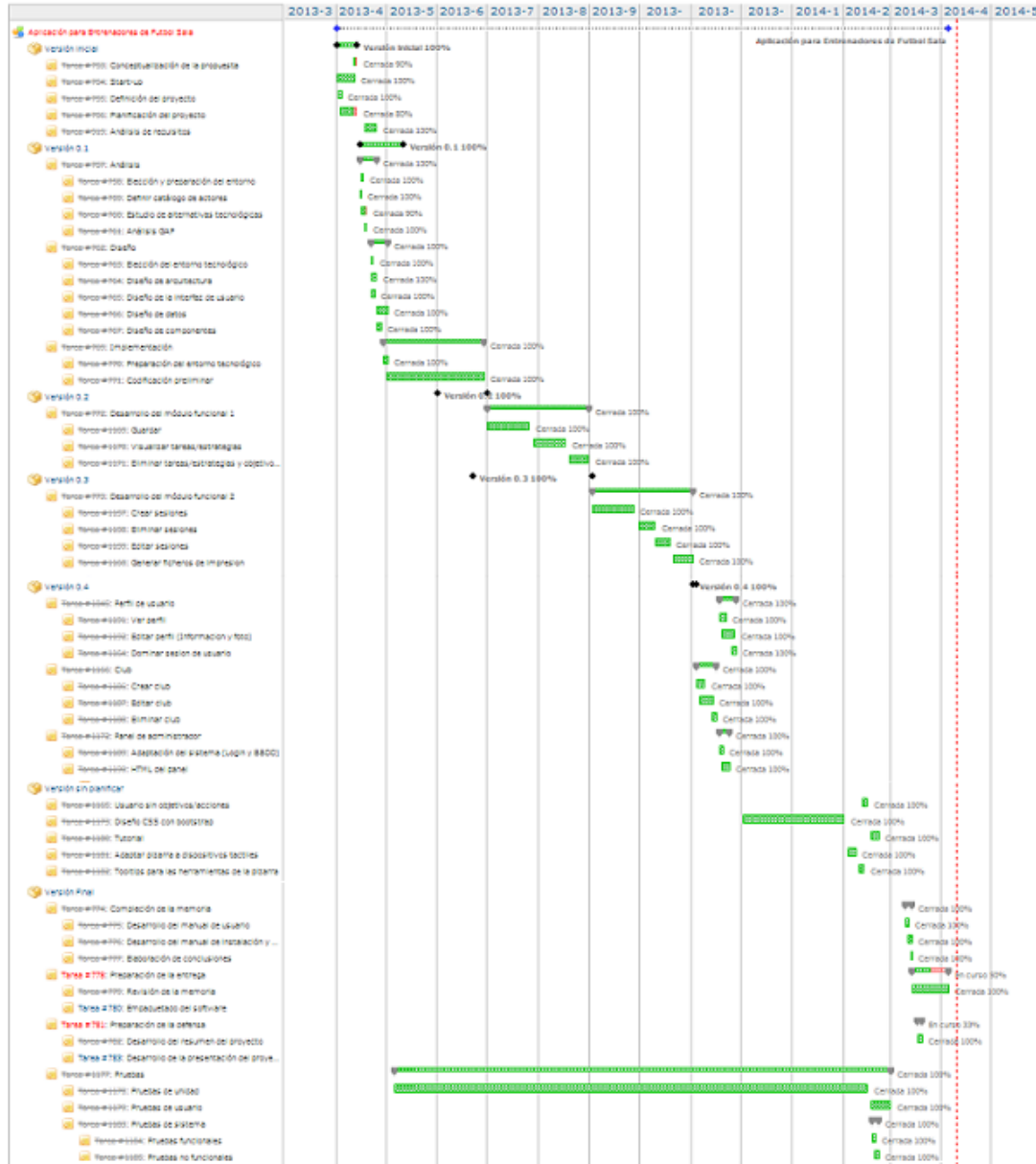
$$\sum \frac{horas}{semana} * n^0 semanas = 25 * 4 + 30 * 4 + 30 * 4 + 30 * 4 = 460 horas$$

Personas/mes estimadas:

$$\sum \frac{p}{m} = 0,625 + 0,75 + 0,75 + 0,75 = 2,875 \text{ personas/mes}$$

La definición del calendario de actividades realizadas durante el proceso de desarrollo del proyecto se puede apreciar en el diagrama de Gantt de la figura 2.3. Este diagrama ha sido obtenido de *redmine* y refleja la realidad acerca de la evolución del proceso.

Figura 2.3: Calendario operativo. Diagrama de Gantt



La estimación de horas de personal se ha recogido a partir del número de horas empleadas al desarrollo del proyecto. Dicha estimación queda registrada en la tabla 2.2. Puede ser

que algunas de estas horas no hayan sido dedicadas directamente a la realización de tareas incluidas al diagrama anterior. Pues, hay que tener en cuenta que se requiere tiempo para actividades como instalación de software, aprendizaje de lenguajes y/o herramientas, etc...

Tabla 2.2: Costes de personal real

Mes	Horas/Semana	Semanas	Personas/Mes (P/M)
Abril	25	4	$\frac{25}{40} * pers/mes = 0,625 * pers/mes$
Mayo	30	4	$\frac{30}{40} * pers/mes = 0,75 * pers/mes$
Junio	30	4	$\frac{30}{40} * pers/mes = 0,75 * pers/mes$
Julio	20	4	$\frac{20}{40} * pers/mes = 0,5 * pers/mes$
Agosto	20	4	$\frac{20}{40} * pers/mes = 0,5 * pers/mes$
Septiembre	30	4	$\frac{30}{40} * pers/mes = 0,75 * pers/mes$
Octubre	30	4	$\frac{30}{40} * pers/mes = 0,75 * pers/mes$
Noviembre	30	4	$\frac{30}{40} * pers/mes = 0,75 * pers/mes$
Diciembre	30	3	$\frac{30}{40} * \frac{3}{4} pers/mes = 0,5625 * pers/mes$
Enero	30	4	$\frac{30}{40} * pers/mes = 0,75 * pers/mes$
Febrero	40	4	$\frac{40}{40} * pers/mes = 1 * pers/mes$
Marzo	40	3	$\frac{40}{40} * \frac{3}{4} pers/mes = 0,75 * pers/mes$
Abril	40	2	$\frac{40}{40} * \frac{2}{4} pers/mes = 0,5 * pers/mes$

Horas reales:

$$\sum \frac{horas}{semana} * n^o semanas = (25 + 30 * 6 + 20 * 2 + 40) * 4 + (30 + 40) * 3 + 40 * 2 = 1430 horas$$

Personas/mes reales:

$$\sum \frac{p}{m} = 0,625 + 0,75 * 7 + 0,5 * 3 + 0,5625 + 1 = 8,9375 personas/mes$$

## 2.3. Organización

Las personas involucradas en este proyecto juegan dos papeles bien diferenciados. El desarrollo del sistema queda en manos de Rafael Gamero del Río y como director del proyecto situamos a Iván Ruiz Rube.

La comunicación entre ambos se realiza principalmente a través de correo electrónico con el objetivo prioritario de resolver dudas acerca del proyecto. No obstante, vía email también se programan citas personales en el despacho del director, para que la comunicación y comprensión entre ambos sea favorable. Otra forma de comunicación, aunque menos directa, es la herramienta *redmine*. Donde el desarrollador, Rafael Gamero del Río, actualiza las tareas, actividades o versiones del proyecto, de modo que el director puede ver el estado cuando le interese. Esta herramienta ofrece además un repositorio donde quedan recogidos los archivos actualizados prácticamente a diario. De este modo, el director del proyecto puede constatar los avances del proyecto.

A nivel hardware, el único recurso utilizado es un ordenador personal. En cuanto al software, se utiliza un servidor *xampp 1.7.3* que consta principalmente de una base de datos *MySQL*, un servidor *Apache* y soporte para lenguajes de script como php y Perl. Por su parte, la implementación se realizará con un editor de textos plano *Notepad++*. Además de esto, empleamos un procesador de textos para la memoria, *LyX*, basado en el lenguaje *Latex* y con filosofía *WYSIWYM* (What You See Is What You Mean), que significa “lo que ves es lo que quieres decir”.

Cabe mencionar que una vez desarrolladas las primeras versiones del software se han subido a un servidor gratuito de hosting, *x10Hosting*. Donde el cliente, Sergio Barroso, ha ido haciendo uso de la aplicación con el objetivo de verificar que satisface sus necesidades.

## 2.4. Costes

El coste total (C) del proyecto supone la suma del coste de los recursos materiales y de los recursos humanos.

$$\textit{Coste total} = \textit{Coste material} + \textit{Coste personal}$$

Los costes materiales ( $C_m$ ) son constituidos por el coste del software ( $C_s$ ) y del hardware ( $C_h$ ). Todos los productos software empleados tienen licencia de código abierto y, por tanto, no suponen costes. El relativo al hardware supone tan solo el precio del ordenador personal del desarrollador, es decir, aproximadamente 580 €.

$$\textit{Coste material} : C_m = C_s + C_h = 0 + 580 = 580 \text{ €}$$

Para el cálculo de costes de personal ( $C_p$ ) consideramos las tablas salariales de la UCA para el personal contratado laboral [CCOO, 2010], que fija el salario de un titulado de grado medio en Informática en 1353.22 euros mensuales.

A partir de los recuentos de horas de trabajo estimadas en la sección 2.2 Planificación del proyecto. De este modo, podremos calcular tanto el coste de personal estimado como el real.

$$\textit{Coste personal} = \textit{salario} * \textit{personas/mes}$$



*Coste personal estimado* :  $C_{pe} = 1353,22 * 2,875 = 3890,51\text{€}$

*Coste personal real* :  $C_{pr} = 1353,22 * 8,9375 = 12094,40\text{€}$

Una vez conocidos estos costes, es posible obtener el coste total estimado y el coste real del proyecto.

*Coste total estimado* :  $C_{pe} + C_m = 3890,51 + 580 = 4470,51\text{€}$

*Coste total real* :  $C_{pr} + C_m = 11417,79 + 580 = 11997,79\text{€}$

## 2.5. Gestión de riesgos

Un riesgo es un evento incierto que, en caso de suceder produce un efecto positivo o negativo sobre los objetivos del proyecto. La gestión de riesgos tiene la función de identificar, estudiar y eliminar las fuentes de riesgo antes de que amenacen el éxito del proyecto. A continuación, se enumerarán un serie de riesgos que pueden afectar a este proyecto:

1. Se pretende desarrollar un software generalizado, a partir de las necesidades de un entrenador en particular, para que pueda usarlo cualquier entrenador de fútbol sala. Existe el riesgo de que el sistema no sea aceptado en líneas generales por los usuarios potenciales.

**Impacto:** Todas las fases podrían verse afectadas, pues el problema radicaría en la definición de requisitos realizada. Por lo tanto, todo el desarrollo en base a estos requisitos puede ser en vano.

**Probabilidad:** La probabilidad de ocurrencia dependerá de las necesidades del público al que va dirigido y de la calidad y fiabilidad del software obtenido.

**Prioridad:** Evitar este riesgo es de máxima prioridad, pues puede significar haber trabajado durante meses en un sistema poco productivo.

**Plan de prevención:** Para prevenir que esto ocurra mostraremos un prototipo a algunos de los posibles usuarios y le haremos conocer las funcionalidades, de modo que podamos conocer sus opiniones.

2. De forma contraria al anterior, la buena aceptación del software por parte de los usuarios puede producir el riesgo de superar la capacidad del servidor.

**Impacto:** Este riesgo afectaría a la fase de explotación, pues se produciría una vez que el software esté en uso.

**Probabilidad:** La probabilidad de ocurrencia dependerá por una parte de la propagación del software entre los usuarios. Por otra parte, dependerá de la supervisión del administrador al servidor.

**Prioridad:** Este riesgo supone alta prioridad, pues significa una mala gestión de los recursos. Los usuarios pueden verse afectados por no tener acceso al sistema y, por tanto, insatisfechos.

Plan de actuación: El modo en que nos enfrentaremos a este riesgo será controlar el volumen de accesos al sistema, controlando la capacidad ocupada del SGBD y la concurrencia de usuarios que acceden. De forma que, si estas cifras ascienden será necesario migrar el sistema a un servidor mas potente.

3. Retraso en la entrega debido a la falta de conocimientos del desarrollador acerca de las herramientas o lenguajes a utilizar durante el proyecto.

Impacto: Este riesgo puede afectar a todas las fases, pues en todas ellas el desarrollador requiere el uso de herramientas que desconoce.

Probabilidad: La probabilidad de ocurrencia es altísima. Puesto que el desarrollador no tiene experiencia con la mayoría de las herramientas que usará.

Prioridad: La prioridad será menor a medida que avance el proyecto. Debido a que en este tipo de situaciones, la curva de aprendizaje es cada vez mayor.

Plan de actuación: En esta ocasión no se puede evitar el riesgo, pues es necesario conocer las distintas herramientas. No obstante, se tratará de realizar un buen proceso de aprendizaje de cada herramienta o lenguaje.

# Parte II

## Desarrollo



## Capítulo 3

# Análisis de requisitos

### 3.1. Catálogo de actores

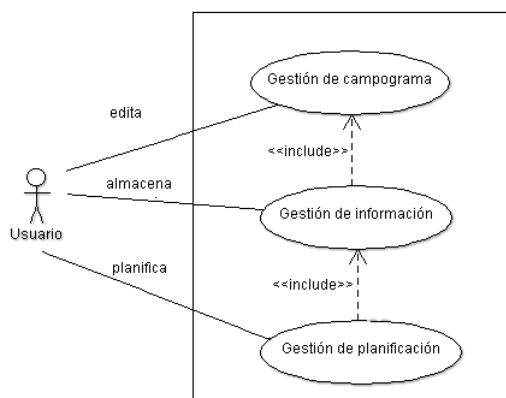
Se pretende desarrollar un sistema que permita a los entrenadores de fútbol sala diseñar e imprimir sesiones de entrenamiento, las cuales estarán compuestas de ejercicios que ellos mismos han de crear, y jugadas de estrategia. De modo que estos son, en principio, los únicos usuarios del software.

Consideramos que el público al que va dirigido este software es adulto y conocedor del entorno en que se centra. Esto facilitará la comprensión de la temática que propondremos. No obstante, este no será motivo para dejar de ofrecer claridad y manejabilidad. Por otra parte, el resultado del uso del software puede llegar a manos de otros componentes del cuerpo técnico o incluso de jugadores y, por esto, los documentos generados deben tratar de ser comprensibles.

### 3.2. Requisitos funcionales

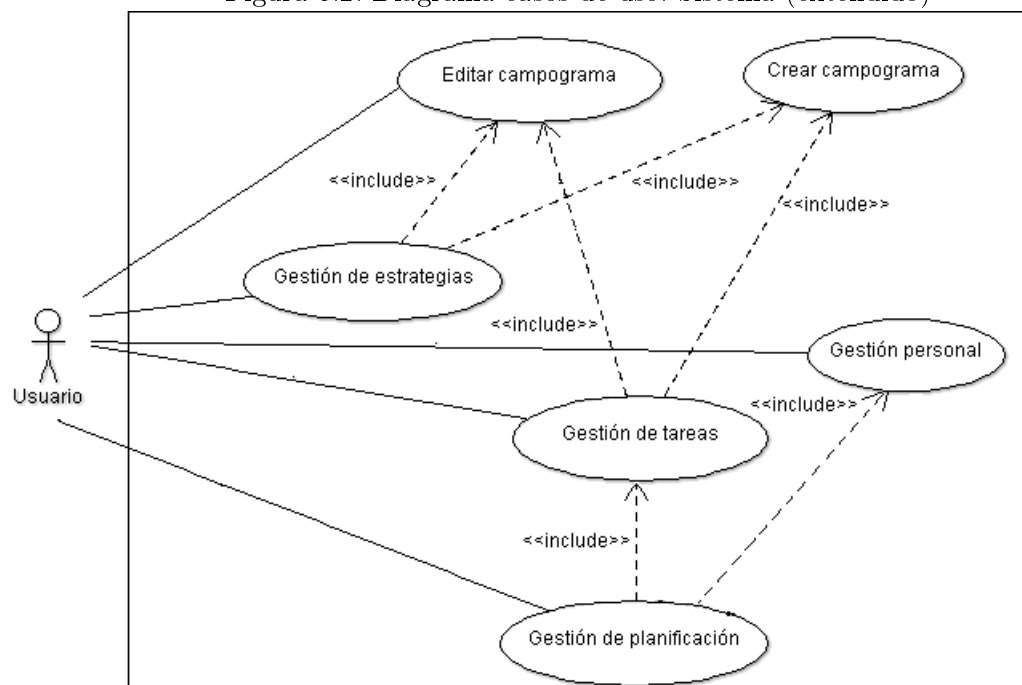
Los requisitos vamos a dividirlos en tres bloques, “Gestión de campograma” donde se editará una pista de fútbol sala para representar tareas o estrategias, “Gestión de información” que tiene como objetivo registrar y mantener la información útil del usuario y, por su parte, la “Gestión de planificación” se encargará de actualizar, mantener y controlar las sesiones de entrenamiento en función al calendario. En la figura 3.1 podemos observar el diagrama de casos de uso del sistema.

Figura 3.1: Diagrama casos de uso: Sistema



Si nos adentramos en el diagrama anterior un poco más, podemos visualizar la funcionalidad del sistema en el de la figura 3.2

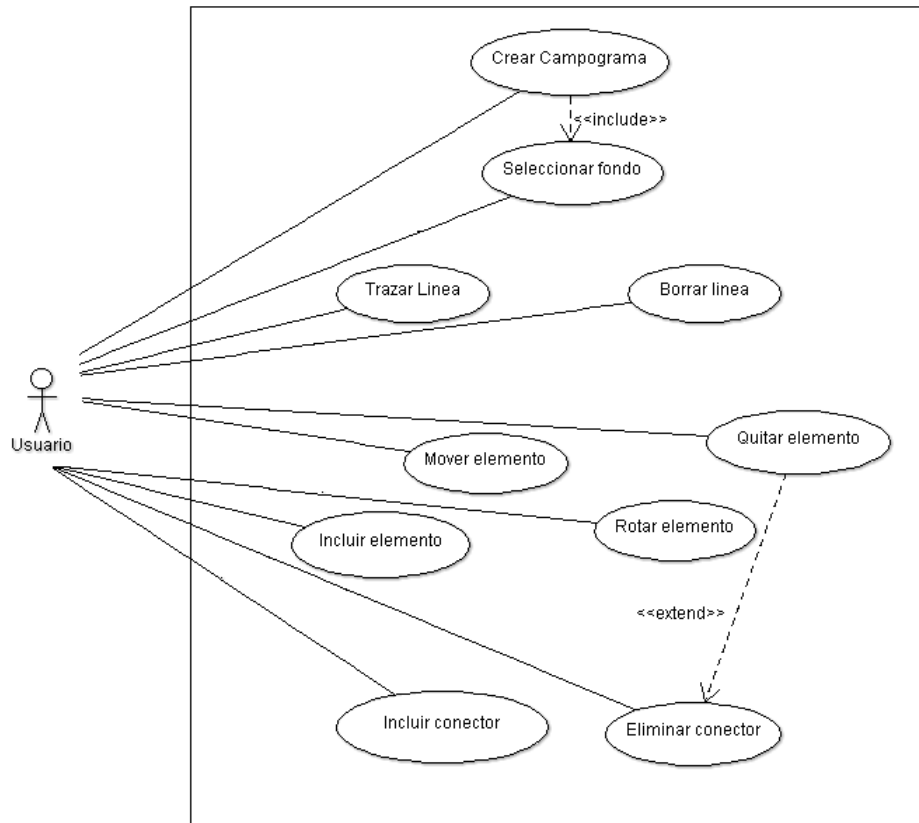
Figura 3.2: Diagrama casos de uso: Sistema (extendido)



## 1. Gestión de campograma

El campograma es un lienzo que simula una pista de fútbol sala, en el que se podrán incluir elementos móviles, girables, conectores y líneas. En la figura 3.3 podemos ver el diagrama de casos de uso relativo a la gestión y edición de campograma.

Figura 3.3: Diagrama casos de uso: Gestión de campograma



## 1.1 Crear campograma

Tabla 3.1: Caso de uso: Crear Campograma

Crear campograma		
Actor	Usuario o Sistema	
Descripción	Se crea un lienzo y herramientas de edición gráfica que representan la pista de fútbol sala	
Precondición		
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicita crear campograma y “1.2.1 Seleccionar fondo”
Paso 2	Sistema	Muestra el campograma y una barra de herramientas para editarlo
Flujo alternativo		

## 1.2 Editar campograma

Se refiere a todas aquellas funciones que añadan, modifiquen o editen elementos del campograma

**1.2.1 Seleccionar tipo pista:** El sistema ofrecerá la representación de dos tipos de pista, la cancha completa o media cancha. El usuario podrá elegir entre estas según convenga al trabajo que simule el campograma.

Tabla 3.2: Caso de uso: Seleccionar pista

<i>Seleccionar pista</i>		
<i>Actor</i>	Usuario	
<i>Descripción</i>	Se elige el fondo del campograma (Pista completa o media pista)	
<i>Precondición</i>	El campograma existe	
<i>Postcondición</i>	Los elementos incluidos no se verán afectados	
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario	Elige el tipo de fondo del campograma
<i>Paso 2</i>	Sistema	Muestra el campograma con el fondo seleccionado
<i>Flujo alternativo</i>		

**1.2.2 Elementos móviles:** Son elementos móviles aquellos que serán representados en el campograma y que podrán ser recolocados. Nombraremos como elementos con estas características a los jugadores, porteros, balones, conos, porterías, etc...

### 1.2.2.1 Incluir elemento

Tabla 3.3: Caso de uso: Incluir elemento

Flujo principal		
Actor	Usuario	
Descripción	Introducir un nuevo elemento al campograma	
Precondición	El campograma existe	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita crear elemento en una posición determinada
Paso 2	Sistema	Muestra el campograma con la representación gráfica del elemento
Flujo alternativo: Posición incorrecta		
Paso 2	Sistema	Muestra el campograma sin el elemento



### 1.2.2.2 Mover elemento

Tabla 3.4: Caso de uso: Mover elemento

Tabla 3.11: Caso de uso: Mover elemento		
Mover elemento		
Actor	Usuario	
Descripción	Modifica la posición de un elemento en el campograma	
Precondición	El campograma y el elemento existen	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita mover elemento a una posición determinada
Paso 2	Sistema	Muestra el campograma con el elemento en la nueva posición
Flujo alternativo: Posición incorrecta		
Paso 2	Sistema	Muestra el campograma sin modificar la posición del elemento

### 1.2.2.3 Eliminar elemento

Tabla 3.5: Caso de uso: Eliminar elemento

Tabla 3.9: Caso de uso: Eliminar elemento		
Eliminar elemento		
Actor	Usuario	
Descripción	Elimina un elemento del campograma	
Precondición	El campograma y el elemento existen	
Postcondición	Los conectores asociados a él también serán eliminados	
Flujo principal		
Paso 1	Usuario	Solicita eliminar un elemento
Paso 2	Sistema	Muestra el campograma sin el elemento ni sus conectores
Flujo alternativo: Conectores asociados al elemento		
Paso 2a	Sistema	“1.2.4.2 Eliminar conector” asociados al elemento. Volver al Paso 2

### 1.2.2.4 Rotar elemento

Algunos de los elementos móviles presentan la posibilidad de ser girados sobre sí mismos en función de la necesidad del usuario. La utilidad de esta funcionalidad radica en que la orientación de estos ítems afecta a la ejecución de la tarea. Como ejemplo de estos podemos considerar porterías o vallas.

Tabla 3.6: Caso de uso: Rotar elemento

Tabla 9.6. Caso de uso: Rotar elemento		
Rotar elemento		
Actor	Usuario	
Descripción	Gira un elemento del campograma	
Precondición	Existe el campograma y el elemento que, además debe presentar la posibilidad de rotar	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita rotar un elemento a la orientación deseada
Paso 2	Sistema	Muestra el campograma con la nueva orientación del elemento
Flujo alternativo		

**1.2.3. Lineas:** Se incluirán líneas para simular los movimientos de los jugadores o del balón. El sistema dispondrá de varios tipos de líneas, de forma que el usuario pueda utilizarlas según necesite. La forma de diferenciar las líneas será el trazo (Continua/Discontinua) y el color.

#### 1.2.3.1 Trazar línea

Tabla 3.7: Caso de uso: Trazar línea

Tabla 5.1: Caso de uso: Trazar línea		
<i>Actor</i>	Usuario	
<i>Descripción</i>	Dibuja una línea en el campograma	
<i>Precondición</i>	Existe el campograma	
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario	Solicita trazar una línea
<i>Paso 2</i>	Usuario	Define el recorrido de la línea
<i>Paso 3</i>	Sistema	Muestra el campograma con la nueva línea
<i>Flujo alternativo:</i> El recorrido sale del campograma		
Paso 3	Sistema	No crea la línea. Muestra el campograma sin representar la línea

### 1.2.3.2 Borrar linea

Tabla 3.8: Caso de uso: Borrar linea  
*Borrar linea*

Tabla 3.3: Caso de uso: Borrar línea		
Actor	Usuario	
Descripción	Borra una línea del campograma	
Precondición	Existe el campograma y la línea	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita borrar una línea
Paso 2	Sistema	Muestra el campograma sin la línea
Flujo alternativo		

### 1.2.3.3 Editar color linea

Tabla 3.9: Caso de uso: Editar color linea  
*Editar color linea*

Tabla 9.9. Caso de uso: Editar color línea		
<i>Editar color línea</i>		
<i>Actor</i>	Usuario	
<i>Descripción</i>	Edita el color del trazo de una línea del campograma	
<i>Precondición</i>	Existe el campograma y la línea	
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario	Solicita cambiar el color de una línea
<i>Paso 2</i>	Sistema	Muestra el campograma con el color de la línea
<i>Flujo alternativo</i>		

**1.2.4 Elementos conectores:** Representan elementos que unen dos jugadores, como pueden ser las gomas elásticas o cuerdas.

#### 1.2.4.1 Incluir conector

Tabla 3.10: Caso de uso: Incluir conector  
*Incluir conector*

Tabla 3.10. Caso de uso. Incluir conector		
Actor	Usuario	
Descripción	Une dos jugadores del campograma	
Precondición	Existe el campograma y los dos jugadores	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita crear conector
Paso 2	Usuario	Selecciona dos jugadores
Paso 3	Sistema	Muestra el campograma con el nuevo conector
Flujo alternativo: Selecciona elemento no válido		
Paso 3	Sistema	No crea el conector. Muestra el campograma sin el conector

#### 1.2.4.2 Eliminar conector

Tabla 3.11: Caso de uso: Eliminar conector

Tabla 5.11: Caso de uso: Eliminar conector		
<i>Eliminar conector</i>		
<i>Actor</i>	Usuario o sistema	
<i>Descripción</i>	Quita el conector del campograma	
<i>Precondición</i>	Existe el campograma y el conector	
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario/Sistema	Solicita eliminar conector
<i>Paso 2</i>	Sistema	Muestra el campograma sin el conector
<i>Flujo alternativo</i>		

#### 1.2.4.3 Editar color conector

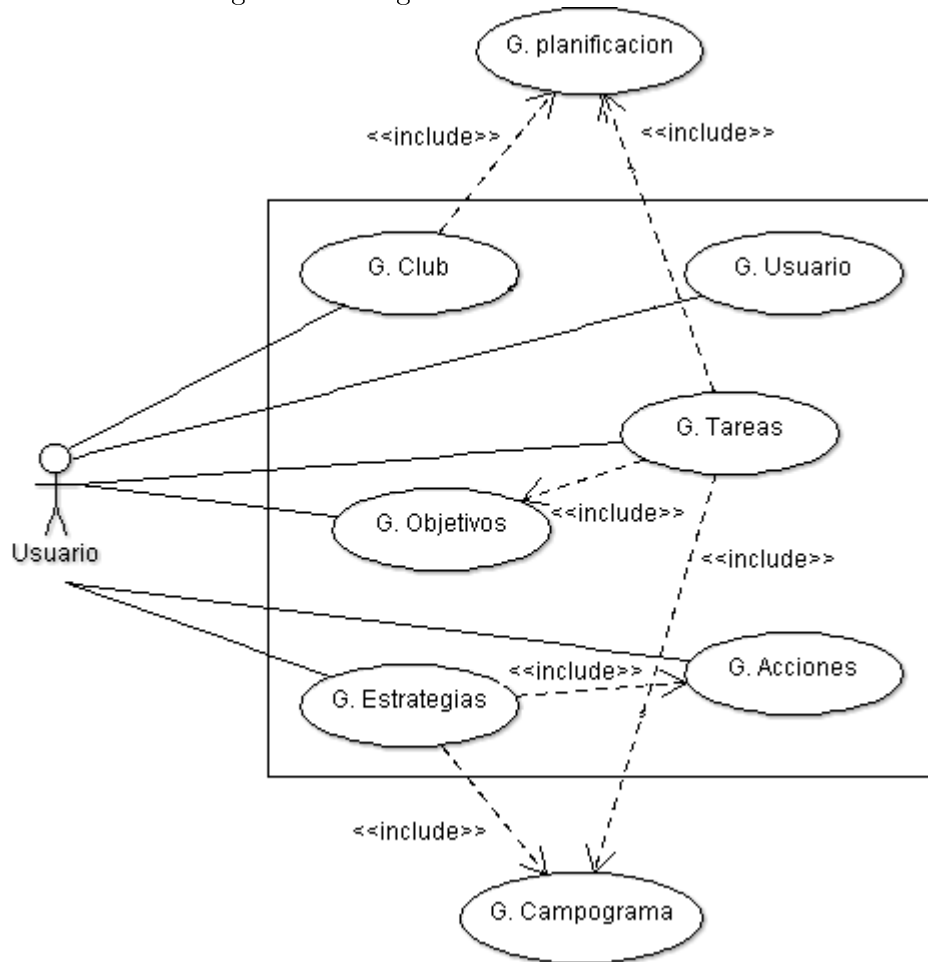
Tabla 3.12: Caso de uso: Editar color conector

Tabla 9.12: Caso de uso: Editar color conector		
Actor	Usuario	
Descripción	Edita el color del trazo de un conector del campograma	
Precondición	Existe el campograma y el conector	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita cambiar el color de un conector
Paso 2	Sistema	Muestra el campograma con el color del conector
Flujo alternativo		

## 2. Gestión de información

La gestión de información se encarga de almacenar y mantener información relevante al usuario (tareas, estrategias, clubes...). De forma que se disponga en buen estado en el momento que se requiera. El siguiente diagrama de caso de uso, figura 3.4, describe el comportamiento de esta sección del sistema.

Figura 3.4: Diagrama casos de uso: Gestión de información



## 2.1. Gestión de tareas

Las tareas son entidades que describen gráfica y textualmente ejercicios que se realizarán durante las sesiones de entrenamiento. Para la descripción gráfica se asociará la tarea a un campograma y para la descripción textual se rellenarán una serie de campos de texto (Nombre, Organización, Desarrollo, Reglas y Periodización).

Las tareas serán clasificadas por objetivos que el propio usuario tendrá que registrar y, por tanto, podrá también editar, eliminar y visualizar.

En la figura 3.5 podemos ver el diagrama de casos de uso de la gestión de tareas.

Figura 3.5: Diagrama casos de uso: Gestión de tareas



Comenzaremos enumerando los casos de uso relacionados con las tareas (registrar, eliminar, editar, visualizar y listar).

### 2.1.1 Registrar tarea

Tabla 3.13: Caso de uso: Registrar tarea

Registrar tarea		
Actor	Usuario	
Descripción	Almacena la descripción gráfica y textual de un ejercicio y lo clasifica	
Precondición		
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita registrar tarea
Paso 2	Sistema	“1.1 Crear campograma”, “2.1.8 Listar objetivos” y muestra campos descripción textual
Paso 3	Usuario	“1.2 Editar campograma”, rellena los campos y selecciona objetivo
Paso 4	Sistema	Muestra confirmación “Tarea guardada”
Flujo alternativo: No existen objetivos		
Paso 2	Sistema	“2.1.6 Registrar objetivo” . Ir al paso 2
Flujo alternativo: Descripción incompleta		
Paso 4	Sistema	Muestra aviso “Descripción incompleta”. Volver a Paso 3

### 2.1.2 Listar tareas

Tabla 3.14: Caso de uso: Listar tareas

Tabla 5.11: Caso de uso: Listar tareas		
Listar tareas		
Actor	Usuario o sistema	
Descripción	Muestra el nombre de todas las tareas	
Precondición		
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicita listar tareas
Paso 2	Sistema	Muestra una lista con los nombres de las tareas ordenadas por objetivos
Flujo alternativo: No existen tareas		
Paso 2	Sistema	Muestra aviso “No existen tareas”

### 2.1.3 Visualizar tarea

Tabla 3.15: Caso de uso: Visualizar tarea  
*Visualizar tarea*

Tabla 3.15: Caso de uso: Visualizar tarea		
Visualizar tarea		
Actor	Usuario o sistema	
Descripción	Muestra la descripción gráfica y textual de un ejercicio	
Precondición	La tarea existe	
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicita visualizar una tarea
Paso 2	Sistema	Muestra la descripción gráfica y textual de la tarea
Flujo alternativo		

### 2.1.4 Eliminar tarea

Tabla 3.16: Caso de uso: Eliminar tarea  
*Eliminar tarea*

Tabla 3.16. Caso de uso: Eliminar tarea		
<i>Eliminar tarea</i>		
<i>Actor</i>	Usuario o sistema	
<i>Descripción</i>	Elimina la descripción gráfica y textual de un ejercicio	
<i>Precondición</i>	La tarea existe	
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario/Sistema	Solicita eliminar una tarea
<i>Paso 2</i>	Sistema	Muestra confirmación “Tarea eliminada”
<i>Flujo alternativo</i>		

### 2.1.5. Editar tarea

Tabla 3.17: Caso de uso: Editar tarea  
*Editar tarea*

Tabla 5.17. Caso de uso: Editar tarea		
<i>Editar tarea</i>		
<i>Actor</i>	Usuario o sistema	
<i>Descripción</i>	Edita la descripción gráfica y/o textual de un ejercicio	
<i>Precondición</i>	La tarea existe	
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 2</i>	Usuario/sistema	Solicita editar una tarea
<i>Paso 3</i>	Sistema	Muestra herramientas gráficas, campos editables y “2.1.8 Listar objetivos”
<i>Paso 4</i>	Usuario	“1.2 Editar campograma” y/o la información
<i>Paso 5</i>	Sistema	Muestra confirmación “Tarea editada”
<i>Flujo alternativo: Descripción incompleta</i>		
Paso 5	Sistema	Muestra aviso “Descripción incompleta”. Volver a Paso 4



A continuación, describiremos los casos de uso relativos a los objetivos (registrar, eliminar, editar y listar)

### 2.1.6 Registrar objetivo

Tabla 3.18: Caso de uso: Registrar objetivo

<i>Registrar objetivo</i>		
<i>Actor</i>	Usuario o sistema	
<i>Descripción</i>	Crea un objetivo nuevo para clasificar las tareas	
<i>Precondición</i>		
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario/Sistema	Solicita registrar objetivo
<i>Paso 2</i>	Sistema	Muestra los campos del objetivo
<i>Paso 3</i>	Usuario	Rellena los campos y envía
<i>Paso 4</i>	Sistema	Muestra confirmación “Objetivo guardado”
<i>Flujo alternativo</i>		

### 2.1.7 Editar objetivo

Tabla 3.19: Caso de uso: Editar objetivo

<i>Editar objetivo</i>		
<i>Actor</i>	Usuario	
<i>Descripción</i>	Modifica la información de un objetivo	
<i>Precondición</i>	El objetivo existe	
<i>Postcondición</i>	Las tareas seguirán asociadas al objetivo	
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario	Solicita editar un objetivo
<i>Paso 2</i>	Sistema	Muestra los campos editables del objetivo
<i>Paso 3</i>	Usuario	Modifica los campos y envía
<i>Paso 4</i>	Sistema	Muestra confirmación “Objetivo editado con éxito”
<i>Flujo alternativo</i>		

### 2.1.8 Listar objetivos

Tabla 3.20: Caso de uso: Listar objetivos

<i>Listar objetivos</i>		
<i>Actor</i>	Usuario o Sistema	
<i>Descripción</i>	Muestra los objetivos del usuario registrados en el sistema	
<i>Precondición</i>		
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario/Sistema	Solicita listar objetivos
<i>Paso 2</i>	Sistema	Muestra una lista con los objetivos del usuario
<i>Flujo alternativo</i>		

### 2.1.9 Eliminar objetivo

Tabla 3.21: Caso de uso: Eliminar objetivo

<i>Eliminar objetivo</i>		
<i>Actor</i>	Usuario	
<i>Descripción</i>	Elimina la información de un objetivo	
<i>Precondición</i>	El objetivo existe	
<i>Postcondición</i>	No dejará tareas sin asociar a un objetivo	
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario	Solicita eliminar objetivo
<i>Paso 2</i>	Sistema	Muestra aviso “Eliminar tareas del objetivo o reclasificar?”
<i>Paso 3a</i>	Usuario	Selecciona Eliminar
<i>Paso 4a</i>	Sistema	“2.1.4 Eliminar tarea” del objetivo
<i>Paso 4a</i>	Sistema	Muestra confirmación “Objetivo eliminado con éxito”
<i>Paso 3b</i>	Usuario	Selecciona Reclasificar
<i>Paso 4b</i>	Sistema	“2.1.8 Listar Objetivos”
<i>Paso 5b</i>	Usuario	Selecciona un objetivo de la lista
<i>Paso 6b</i>	Sistema	“2.1.5 Editar tarea” para las clasificadas (Objetivo)
<i>Paso 7b</i>	Sistema	Muestra confirmación “Tareas reclasificadas”
<i>Flujo alternativo: Objetivo sin tareas</i>		
<i>Paso 3</i>	Sistema	Muestra confirmación “Objetivo eliminado con éxito”

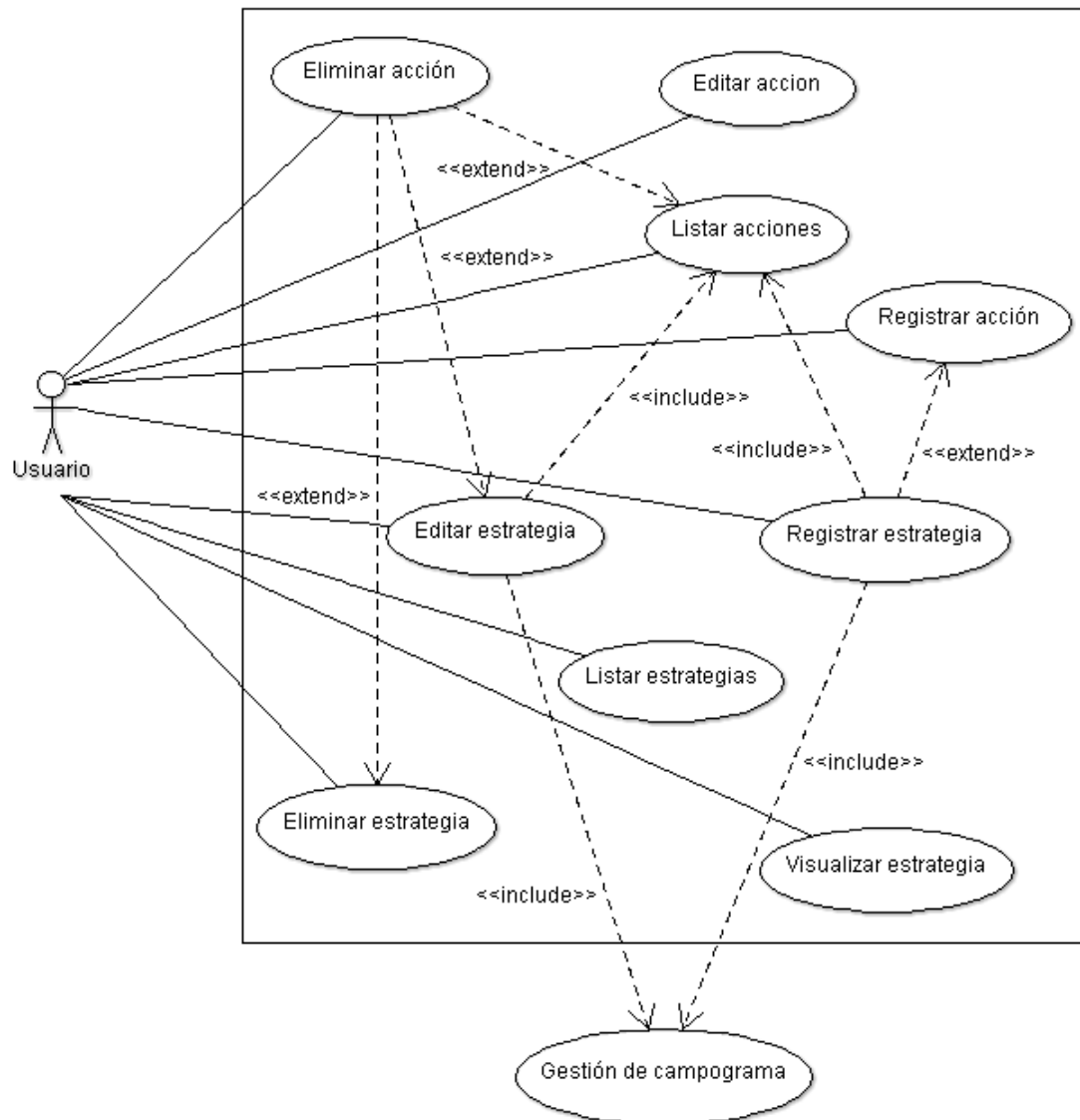
## 2.2. Gestión de estrategias

Las estrategias son entidades que describen gráfica y textualmente jugadas que el entrenador hará conocer a la plantilla. A diferencia de las tareas, no podrán ser incluidas en las sesiones de entrenamiento. Además de esto, también se diferencian en los campos de la descripción textual, que serán (nombre y desarrollo). Para la descripción gráfica se hará uso de un campograma, del mismo modo que las tareas. En la figura 3.6 podemos observar el diagrama de casos de uso de la gestión de estrategias.

Las estrategias serán clasificadas por acciones que el propio usuario definirá en función de sus necesidades. Por tanto, podrá eliminar las acciones, editarlas y visualizarlas.

NOTA: Las estrategias no se incluyen en sesiones de entrenamiento. Si un entrenador pretende trabajar durante una sesión las jugadas de estrategia, tendrá que diseñar una tarea que represente el ejercicio para ensayarlas.

Figura 3.6: Diagrama casos de uso: Gestión de estrategias



Al igual que en la gestión de tareas, comenzaremos describiendo los casos de uso de las estrategias.

### 2.2.1 Registrar estrategia

Tabla 3.22: Caso de uso: Registrar estrategia

Registrar estrategia		
Actor	Usuario	
Descripción	Almacena la descripción gráfica y textual de una jugada de estrategia	
Precondición		
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita registrar estrategia
Paso 2	Sistema	“1.1 Crear campograma”, “2.2.8 Listar acciones” y muestra campos descripción textual
Paso 3	Usuario	“1.2 Editar campograma”, selecciona acción y rellena los campos
Paso 4	Sistema	Muestra confirmación “Estrategia guardada”
Flujo alternativo: No existen acciones		
Paso 2	Sistema	“2.2.6 Registrar acción” . Ir al paso 2
Flujo alternativo: Descripción incompleta		
Paso 4	Sistema	Muestra aviso “Descripción incompleta”. Volver a Paso 3

### 2.2.2 Listar estrategias

Tabla 3.23: Caso de uso: Listar estrategias

<i>Listar estrategias</i>		
<i>Actor</i>	Usuario o sistema	
<i>Descripción</i>	Muestra el nombre de todas las jugadas de estrategia	
<i>Precondición</i>		
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario/Sistema	Solicita listar estrategias
<i>Paso 2</i>	Sistema	Muestra una lista con los nombres de las estrategias
<i>Flujo alternativo: No existen estrategias</i>		
Paso 2	Sistema	Muestra aviso “No existen estrategias”

### 2.2.3 Visualizar estrategia

Tabla 3.24: Caso de uso: Visualizar estrategia

Visualizar estrategia		
Actor	Usuario	
Descripción	Muestra la descripción gráfica y textual de una jugada de estrategia	
Precondición	La estrategia existe	
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicita visualizar una estrategia
Paso 2	Sistema	Muestra la descripción gráfica y textual de la estrategia
Flujo alternativo		

### 2.2.4 Eliminar estrategia

Tabla 3.25: Caso de uso: Eliminar estrategia

<i>Eliminar estrategia</i>		
<i>Actor</i>	Usuario	
<i>Descripción</i>	Elimina la descripción gráfica y textual de una jugada de estrategia	
<i>Precondición</i>	La estrategia existe	
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario	Solicita eliminar estrategia
<i>Paso 2</i>	Sistema	Muestra confirmación “Estrategia eliminada”
<i>Flujo alternativo</i>		

### 2.2.5. Editar estrategia

Tabla 3.26: Caso de uso: Editar estrategia

Tabla 5.2.6. Caso de uso: Editar estrategia		
Editar estrategia		
Actor	Usuario o sistema	
Descripción	Edita la descripción gráfica y/o textual de una jugada de estrategia	
Precondición	La estrategia existe	
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicita editar una estrategia
Paso 2	Sistema	Muestra herramientas gráficas, campos editables y '2.2.8 Listar acciones"
Paso 3	Usuario	'1.2 Editar campograma' y/o la información
Paso 4	Sistema	Muestra confirmación "Estrategia editada"
Flujo alternativo: Descripción textual incompleta		
Paso 4	Sistema	Muestra aviso "Descripción incompleta". Volver a Paso 3

Y, del mismo modo, a partir de aquí definiremos los casos de uso relacionados con las acciones.

### 2.2.6 Registrar acción

Tabla 3.27: Caso de uso: Registrar acción

Registrar acción		
Actor	Usuario o sistema	
Descripción	Crea una acción nuevo para clasificar las estrategias	
Precondición		
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicita registrar acción
Paso 2	Sistema	Muestra los campos de la acción
Paso 3	Usuario	Rellena los campos y envía
Paso 4	Sistema	Muestra confirmación “Acción guardada con éxito”
Flujo alternativo		

### 2.2.7 Editar acción

Tabla 3.28: Caso de uso: Editar acción

Tabla 5. Caso de uso: Editar acción		
Editar acción		
Actor	Usuario	
Descripción	Modifica la información de una acción	
Precondición	La acción existe	
Postcondición	Las estrategias seguirán asociadas a la acción	
Flujo principal		
Paso 1	Usuario	Solicita editar una acción
Paso 2	Sistema	Muestra los campos editables de la acción
Paso 3	Usuario	Modifica los campos y envía
Paso 4	Sistema	Muestra confirmación “Acción editada con éxito”
Flujo alternativo		

### 2.2.8 Listar acciones

Tabla 3.29: Caso de uso: Listar acciones

Tabla 3.23: Caso de uso: Listar acciones		
Listar acciones		
Actor	Usuario o Sistema	
Descripción	Muestra las acciones del usuario registrados en el sistema	
Precondición		
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicita listar acciones
Paso 2	Sistema	Muestra una lista con las acciones del usuario
Flujo alternativo		

### 2.2.9 Eliminar acción

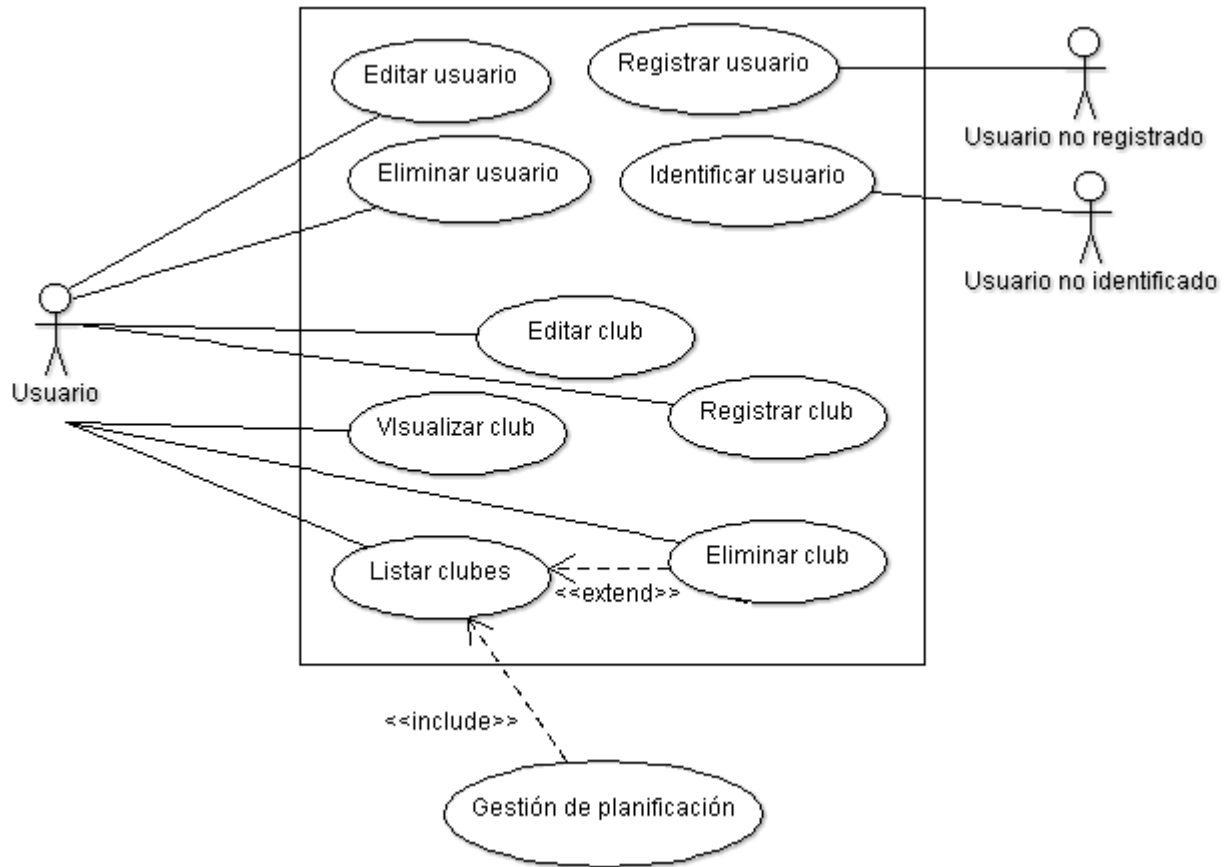
Tabla 3.30: Caso de uso: Eliminar acción

Tabla 3.38: Caso de uso: Eliminar acción		
Actor	Usuario	
Descripción	Elimina la información de una acción	
Precondición	La acción existe	
Postcondición	No dejará estrategias sin asociar a una acción	
Flujo principal		
Paso 1	Usuario	Solicita eliminar acción
Paso 2	Sistema	Muestra aviso “Eliminar estrategias de la acción o reclasificar?”
Paso 3a	Usuario	Selecciona Eliminar
Paso 4a	Sistema	“2.2.4 Eliminar estrategia” para las asociadas
Paso 5a	Sistema	Muestra confirmación “Acción eliminada con éxito”
Paso 3b	Usuario	Selecciona Reclasificar
Paso 4b	Sistema	“2.2.8 Listar Acciones”
Paso 5b	Usuario	Selecciona una acción de la lista
Paso 6b	Sistema	“2.2.5 Editar estrategia” para las asociadas (Cambia acción)
Paso 7b	Sistema	Muestra confirmación “Estrategias reclasificadas”
Flujo alternativo: Acción sin estrategias		
Paso 3	Sistema	Muestra confirmación “Acción eliminada con éxito”

## 2.3. Gestión personal

La información propia del usuario y de los clubes con los que trabaja se gestionará según este apartado. En la figura 3.7 podemos ver el diagrama de casos de uso de la gestión personal.

Figura 3.7: Diagrama casos de uso: Gestión personal



**2.3.1. Gestión de usuarios:** El sistema permite dar de alta a nuevos usuarios y a los usuarios registrados editar su propia información y darse de baja.



### 2.3.1.1 Registrar usuario

Tabla 3.31: Caso de uso: Registrar usuario

Registrar usuario		
<i>Actor</i>	Usuario no registrado	
<i>Descripción</i>	Crea un nuevo usuario del sistema	
<i>Precondición</i>	El usuario no esta registrado o inactivo	
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario no registrado	Solicita registrarse en el sistema
<i>Paso 2</i>	Sistema	Muestra los campos de información para el registro
<i>Paso 3</i>	Usuario no registrado	Rellena los campos y envía
<i>Paso 4</i>	Sistema	Muestra confirmación “Usuario registrado con éxito”
<i>Flujo alternativo: Información de registro incompleta o incorrecta</i>		
<i>Paso 4a</i>	Sistema	Muestra aviso “Información no válida”. Volver a Paso 3
<i>Flujo alternativo: Usuario registrado</i>		
<i>Paso 4b</i>	Sistema	Muestra aviso “Existe usuario registrado en el sistema”

### 2.3.1.2 Identificar usuario

Tabla 3.32: Caso de uso: Identificar usuario

Tabla 3.32: Caso de uso: Identificar usuario		
<i>Actor</i>	Usuario no identificado	
<i>Descripción</i>	Identifica a un usuario que pretende acceder al sistema	
<i>Precondición</i>		
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario no identificado	Solicita acceder a la aplicación
<i>Paso 2</i>	Sistema	Muestra campos para introducir el email y la contraseña
<i>Paso 3</i>	Usuario no identificado	Introduce nombre de usuario, contraseña y envía
<i>Paso 4</i>	Sistema	Permite el acceso al usuario a la aplicación.
<i>Flujo alternativo: Datos incorrectos</i>		
<i>Paso 4</i>	Sistema	Muestra aviso “Información invalida”. Volver a Paso 3

### 2.3.1.3 Eliminar usuario

Tabla 3.33: Caso de uso: Eliminar usuario  
Eliminar usuario

Eliminar usuario		
Actor	Usuario	
Descripción	Elimina la cuenta de un usuario del sistema	
Precondición	El usuario esta registrado	
Postcondición	El sistema mantendrá la información del usuario	
Flujo principal		
Paso 1	Usuario	Solicita darse de baja en el sistema
Paso 2	Sistema	Muestra un campo para introducir la contraseña
Paso 3	Usuario	Introduce la contraseña y envía
Paso 4	Sistema	Muestra confirmación “Usuario eliminado” y cierra
Flujo alternativo: Contraseña incorrecta		
Paso 4	Sistema	Muestra aviso “Usuario no eliminado”. Volver a Paso 3 (3 veces)

### 2.3.1.4 Editar usuario

Tabla 3.34: Caso de uso: Editar usuario  
Editar usuario

<div> <div>Tabla 3.3.1. Caso de uso: Editar usuario</div> <div>Editar usuario</div> </div>		
<i>Actor</i>	Usuario	
<i>Descripción</i>	Edita la información del propio usuario	
<i>Precondición</i>	El usuario esta registrado	
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario	Solicita editar información
<i>Paso 2</i>	Sistema	Muestra los campos editables
<i>Paso 3</i>	Usuario	Edita la información necesaria y envía
<i>Paso 4</i>	Sistema	Muestra confirmación “Usuario editado con éxito”
<i>Flujo alternativo</i>		

**2.3.2. Gestión de clubes:** Un usuario puede registrar varios clubes, ya que puede dirigirlos al mismo tiempo, o bien, haber cambiado de club una vez registrado en el sistema. En este último caso, puede interesar conservar la información de las sesiones realizadas con el club anterior.

### 2.3.2.1 Registrar club

Tabla 3.35: Caso de uso: Registrar club

Registrar club		
<i>Actor</i>	Usuario	
<i>Descripción</i>	Crea un nuevo club	
<i>Precondición</i>		
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario	Solicita registrar club
<i>Paso 2</i>	Sistema	Muestra los campos de información para el registro
<i>Paso 3</i>	Usuario	Rellena los campos y envía
<i>Paso 4</i>	Sistema	Muestra confirmación “Club registrado con éxito”
<i>Flujo alternativo</i>		
<i>Paso 4</i>	Sistema	Muestra aviso “Información de club incompleta”. Volver al 3

### 2.3.2.2 Listar clubes

Tabla 3.36: Caso de uso: Listar clubes

Tabla 3.3.1. Usos de UML: Listar clubes		
<i>Listar clubes</i>		
<i>Actor</i>	Usuario o sistema	
<i>Descripción</i>	Muestra el nombre de todos los clubes	
<i>Precondición</i>		
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario/Sistema	Solicita listar clubes
<i>Paso 2</i>	Sistema	Muestra una lista con los nombres de los clubes
<i>Flujo alternativo: No hay clubes registrados</i>		
<i>Paso 2</i>	Sistema	Muestra aviso “No existen clubes”

### 2.3.2.3 Visualizar club

Tabla 3.37: Caso de uso: Visualizar club

Tabla 3.3.1: Caso 4.3.1: Visualizar club		
Visualizar club		
Actor	Usuario o sistema	
Descripción	Muestra la información relativa a un club (Nombre, escudo, historial...)	
Precondición	El club existe	
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicitar visualizar un club
Paso 2	Sistema	Muestra la información del club
Flujo alternativo		

#### 2.3.2.4 Eliminar club

Tabla 3.38: Caso de uso: Eliminar club  
*Eliminar club*

Paso 3.35: Caso de uso: Eliminar club		
Eliminar club		
Actor	Usuario	
Descripción	Elimina la información de un club	
Precondición	El club existe	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita eliminar un club
Paso 2	Sistema	Muestra confirmación “Club eliminado”
Flujo alternativo		

#### 2.3.2.5 Editar club

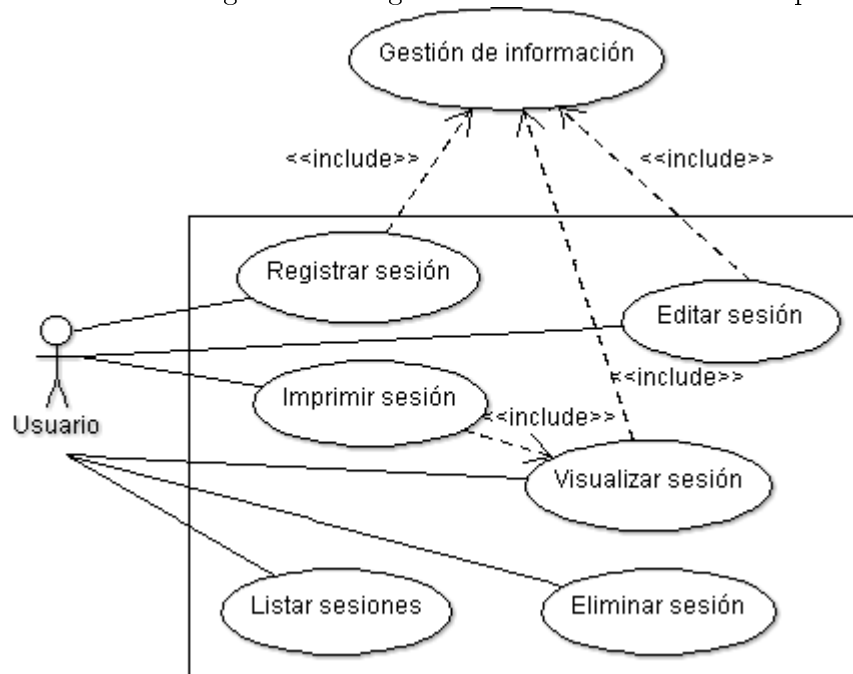
Tabla 3.39: Caso de uso: Editar club  
*Editar club*

Tabla 5.55. Caso de uso: Editar club		
Actor	Usuario	
Descripción	Edita la información de un club	
Precondición	El club existe	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita editar un club
Paso 2	Sistema	Muestra la información editable del club
Paso 3	Usuario	Edita la información necesaria y envía
Paso 4	Sistema	Muestra confirmación “Club editado con éxito”
Flujo alternativo: Información incompleta		
Paso 4	Sistema	Muestra aviso “Información de club incompleta”. Volver al 3

### 3. Gestión de planificación

La gestión de planificación gira en torno a las sesiones de entrenamiento. Dichas sesiones están compuestas de cierta información que describa la sesión (fecha, hora, lugar, etc...), el club que la realizará y un conjunto de tareas que representan las que serán realizadas físicamente. Este apartado queda descrito en el diagrama de casos de uso de la figura 3.8.

Figura 3.8: Diagrama casos de uso: Gestión de planificación



### 3.1 Registrar sesión

Tabla 3.40: Caso de uso: Registrar sesión

Registrar sesión		
Actor	Usuario	
Descripción	Crea una nueva sesión de entrenamiento	
Precondición		
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita registrar sesión
Paso 2	Sistema	Muestra los campos de información, “2.3.2.2 Listar clubes” y ‘2.1.2 Listar tareas’
Paso 3	Usuario	Rellena los campos y elige club
Paso 4	Usuario	“2.1.3 Visualizar tarea” antes de añadirlas a la sesión y envía
Paso 5	Sistema	Muestra confirmación “Sesión registrada con éxito”
Flujo alternativo: Información de sesión incompleta		
Paso 5	Sistema	Muestra aviso “Información de sesión incompleta”. Volver a Paso 3
Flujo alternativo		

### 3.2 Listar sesiones

Tabla 3.41: Caso de uso: Listar sesiones

Tabla 5.11. Caso de uso: Listar sesiones		
Listar sesiones		
Actor	Usuario o sistema	
Descripción	Muestra en el calendario los días para los que se han planificado sesiones	
Precondición		
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicita listar sesiones
Paso 2	Sistema	Muestra el calendario marcando los días en los que se han creado sesiones
Flujo alternativo:		

### 3.3 Visualizar sesión

Tabla 3.42: Caso de uso: Visualizar sesión

Tabla 3.12: Caso de uso: Visualizar sesión		
Visualizar sesión		
Actor	Usuario	
Descripción	Muestra la información de la sesión y las tareas que la componen	
Precondición	La sesión existe	
Postcondición		
Flujo principal		
Paso 1	Usuario/Sistema	Solicita visualizar una sesión
Paso 2	Sistema	Muestra información de la sesión y
		“2.1.3 Visualizar tarea” para las asociadas
Flujo alternativo		

### 3.4 Editar sesión

Tabla 3.43: Caso de uso: Editar sesión

Tabla 5.15: Caso de uso: Editar sesión		
Editar sesión		
Actor	Usuario	
Descripción	Permite modificar la información de la sesión y el conjunto de tareas	
Precondición	La sesión existe	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita editar una sesión
Paso 2	Sistema	Muestra la información editable y la lista de tareas incluidas
Paso 3	Sistema	“2.1.2 Listar tareas” y “2.3.2.2 Listar clubes”
Paso 4	Usuario	Edita la información y/o el conjunto de tareas de la sesión
Paso 5	Sistema	Muestra confirmación “Sesión editada con éxito”
Flujo alternativo: Información de sesión incompleta		
Paso 5	Sistema	Muestra aviso “Información de sesión incompleta”. Volver al 3
Flujo alternativo		

### 3.5 Eliminar sesión

Tabla 3.44: Caso de uso: Eliminar sesión

Tabla 9.11. Caso de uso: Eliminar sesión		
Eliminar sesión		
Actor	Usuario	
Descripción	Elimina la sesión del calendario	
Precondición	La sesión existe	
Postcondición		
Flujo principal		
Paso 1	Usuario	Solicita eliminar una sesión
Paso 2	Sistema	Muestra confirmación “Sesión eliminada”
Flujo alternativo		

### 3.6 Imprimir sesión

Este es el objetivo principal del software. Todo el trabajo por mi parte como desarrollador y del usuario como entrenador tiene como fin imprimir la plantilla que sirva de guía al responsable del entrenamiento para dirigirlo.

Tabla 3.45: Caso de uso: Imprimir sesión

Imprimir sesión		
<i>Actor</i>	Usuario	
<i>Descripción</i>	Genera un fichero imprimible con la información de la sesión	
<i>Precondición</i>	La sesión existe	
<i>Postcondición</i>		
<i>Flujo principal</i>		
<i>Paso 1</i>	Usuario	Solicita imprimir una sesión
<i>Paso 2</i>	Sistema	“3.3 Visualizar sesión” y generar fichero de impresión
<i>Flujo alternativo</i>		



### 3.3. Requisitos de información

Los requisitos de información se ven reflejados en la figura 3.9 generado con el software *ArgoUML*. No obstante, a continuación se describen cada una de las clases que aparecen en el modelo:

**Usuario** Almacena la información relativa a cada usuario del sistema. Contiene por un lado campos de información introducida por el usuario tales como su email, nombre, foto, etc. Por otro lado, información de control como un identificador de usuario o la fecha de registro, etc.

**Tarea** Contiene la estructura necesaria para mantener el nombre y la descripción textual de las tareas que diseñen los usuarios.

**Objetivo** Clasifica las tareas que el usuario introduzca en el sistema. Es decir, un objetivo debe ser representativo del fin por el que se realiza una tarea.

**Estrategia** Una instancia de esta clase contendrá el nombre y la descripción textual de una estrategia.

**Accion** Agrupa las estrategias que diseñe el usuario en función del tipo de jugada que se trate.

**Club** Mantiene información relativa a los clubes de los usuarios.

**Sesion** Cada instancia representa una sesión de entrenamiento. Por tanto, contendrá información de ella como fecha, hora, lugar, club para el que se diseña, etc... Además de esto, tendrá que mantener relación con el conjunto las tareas que se realizarán durante esta sesión.

**Campograma** Contiene información de la descripción gráfica de una tarea o estrategia. Para ello almacena el tipo de fondo de pista y agrupará los elementos que el usuario ha incluido en dicho campograma.

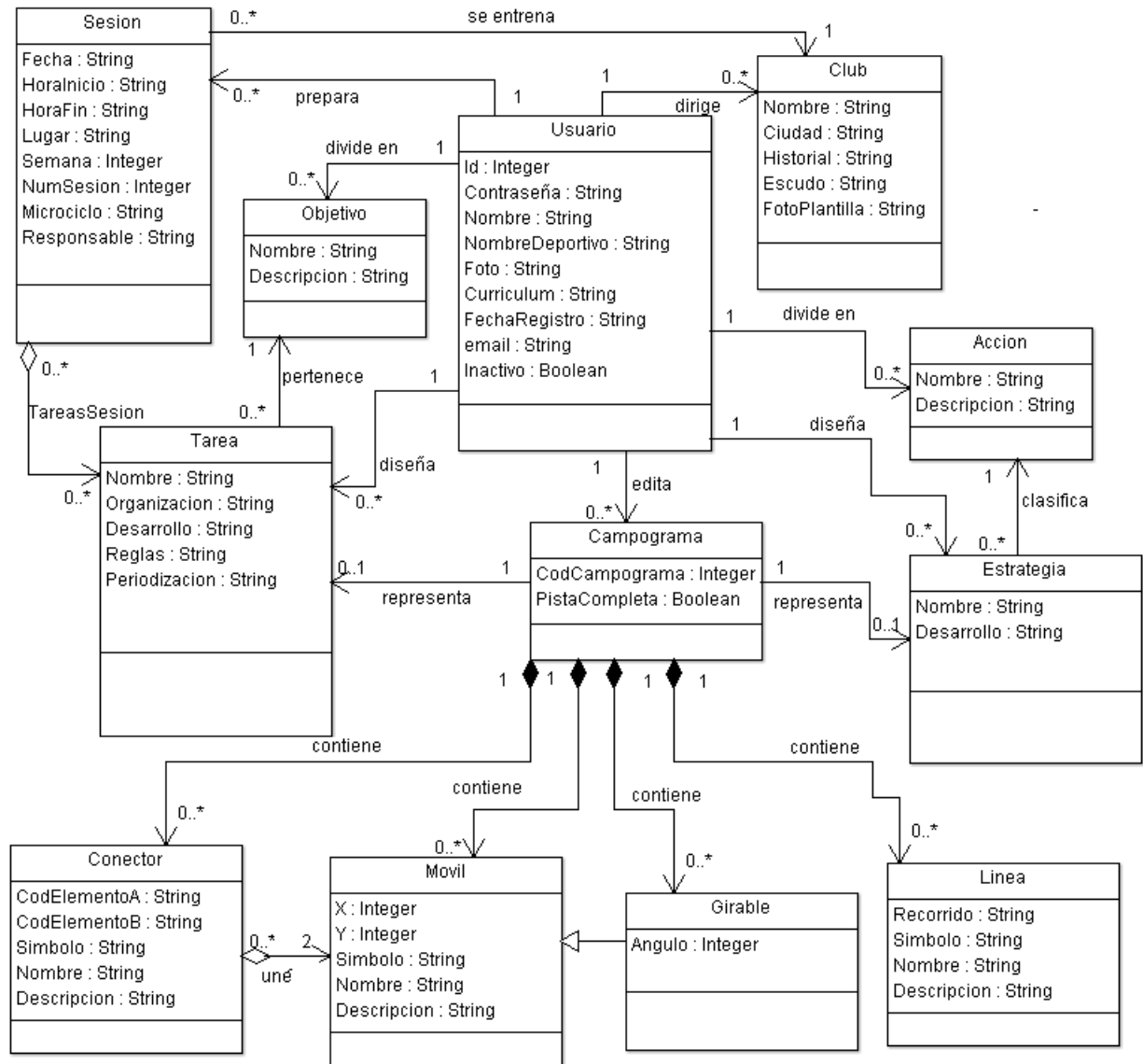
**Movil** Almacena la información de los elementos móviles del campograma como su posición, descripción del tipo, etc...

**Girable** Esta clase es una especialización de la anterior. Contiene el ángulo de giro que tiene el elemento en el campograma.

**Conector** Contiene información del tipo de elemento y de los móviles que conecta.

**Linea** Registra información del tipo de linea y del trazo que conforma en el campograma.

Figura 3.9: Diagrama conceptual de clases



### 3.4. Requisitos no funcionales

**Portabilidad** Se ofrecerá un software multiplataforma. De modo que, la aplicación tratará de ser compatible con los navegadores mas populares. Entre los que podemos mencionar *Mozilla Firefox*, *Internet Explorer*, *Google Chrome*, *Safari*, *Android Browser*. Ademas se pretende proporcionar un sistema adaptable a distintos tipos de dispositivos tales como PCs, tablets o smartphones.

**Seguridad** se reflejará en dos aspectos:

- Login de usuario: Cada usuario que pretenda acceder al sistema debe tener su propia cuenta que contendrá un identificador y una clave, la cual será encriptada para almacenarla.
- Acceso restringido: Un usuario solo podrá acceder a los recursos que le pertenecen.

**Auditoria** Se perseguirán los objetivos y los planes marcados en el análisis del sistema.

**Fiabilidad** Realizaremos una buena implementación de tecnologías adecuadas y, por otra parte, proponiendo un uso sencillo y ofreciendo ventanas de confirmación conseguiremos un sistema fiable.

**Extensibilidad** Se pretende desarrollar un software extensible. De modo que, se puedan ofrecer nuevas funcionalidades al tiempo que vayan surgiendo necesidades a los usuarios del sistema.

**Escalabilidad** Se prevé que la cantidad de usuarios sea creciente. Y, por tanto, se pretende ofrecer la misma calidad de servicio sea cual sea el volumen de usuarios del sistema.

**Rendimiento** Trataremos de realizar un sistema que aproveche los recursos. Por una parte, se tratará de organizar y optimizar los ficheros propios de código. Y, por otra parte, utilizaremos bibliotecas especializadas en las funcionalidades que lo permitan para asegurarnos el máximo rendimiento.

**Interfaz** Ofrecerá un diseño sencillo, de fácil comprensión para proporcionar buena manejabilidad y comprensión del software.

### 3.5. Reglas de negocio

Este proyecto no va dirigido a un cliente en concreto pero la propuesta proviene de un entrenador. Por lo que nos ajustaremos a las preferencias de este. Consideramos algunas reglas que hacen extensible el sistema.

- Una estrategia no puede ser incluida en una sesión. Es decir, si un entrenador pretende trabajar las jugadas de estrategia durante la sesión, deberá crear una “Tarea” que permita ensayar las estrategias deseadas. No obstante, desde la visualización de una “Estrategia” se podrá crear una tarea, incluyendo únicamente la descripción textual necesaria.

- Un entrenador puede estar al frente de varios equipos al mismo tiempo. Este es el caso del autor de la propuesta, que dirige un club al mismo tiempo que es seleccionador del equipo de la universidad. Por esta razón, deberá registrar los dos “clubes” y así podrá diseñar sesiones de entrenamiento para ambos.
- Consideramos oportuno no destruir la información de un usuario que desee darse de baja del sistema. Pues, es posible que en algún momento intente recuperar su información.
- Pueden crearse sesiones sin tareas aunque no sea habitual. Estas pueden representar sesiones de reunión.

### 3.6. Estudio de las alternativas tecnológicas

A la hora de desarrollar un proyecto se nos presentan dos opciones, usar software base o realizar un desarrollo propio.

#### Software base

En primer lugar, queremos destacar que el autor de la propuesta no usa ningún software para planificar sus entrenamientos. Una de las razones es que entre los sistemas disponibles no considera que ninguno se ajuste a sus necesidades reales.

Además de esto, deseamos la opción de basarnos en un software, pues no encontramos licencias libres en el mercado de aplicaciones enfocadas a entrenadores.

#### Desarrollo de una propia aplicación

En este momento, debido a las facilidades que ofrece tanto de programación, instalación/mantenimiento y acceso, se encuentran en auge las aplicaciones web. Son sistemas a los que se accede con suma facilidad, independientes del sistema operativo, pues solo requieren un navegador, y sencillas de distribuir e instalar. Es por esto que nos decantamos por enfocar la realización de la aplicación en este ámbito. Por tanto, tendremos que tomar decisiones en función al motor de base de datos y, por otro lado, tendremos que elegir los lenguajes que vamos a utilizar y si nos basaremos en el alguna plataforma tecnológica de desarrollo.

#### Plataformas tecnológicas

Entre las plataformas de desarrollo actuales mencionaremos las siguientes:

**.NET**<sup>1</sup> es una plataforma de desarrollo y ejecución de aplicaciones; es decir, proporciona los elementos necesarios para el desarrollo de aplicaciones de software, y todos los mecanismos de seguridad y eficiencia para asegurar su óptima ejecución.

*.NET* podría considerarse una respuesta de *Microsoft* al creciente mercado de los negocios en entornos Web, como competencia a la plataforma *Java* de *Oracle Corporation* y a los diversos framework de desarrollo web basados en *PHP*.

---

<sup>1</sup>.NET: <http://www.microsoft.com/net>

**Java EE**<sup>2</sup> también conocida como *JEE*, Su nombre completo es *Java Platform, Enterprise Edition* y, anteriormente era conocido como *J2EE*. Se trata de una plataforma de desarrollo y ejecución de aplicaciones bajo el lenguaje *Java*. Permite arquitecturas de varias capas distribuidas y se basa en componentes de software modulares que son ejecutados en un servidor de aplicaciones. Por otra parte, es también considerada informalmente un estándar, ya que los proveedores deben cumplir ciertos requisitos de conformidad para declarar que sus productos son acordes a *Java EE*.

## Sistema de gestión de base de datos (SGBD)

A continuación enumeraremos algunos de los SGBD mas comunes en el mercado actualmente:

**SQLite**<sup>3</sup> Es un SGBD relacional muy ligero. No requiere la instalación de ningún tipo de servidor, sino que basa su funcionamiento en una serie de archivos en el sistema de ficheros y una pequeña librería que implementa las funcionalidades de la BBDD.

Es un proyecto de dominio público y tiene soporte desde una gran variedad de lenguajes de servidor.

Debido a su pequeño tamaño, es la elección para los desarrollos de aplicaciones móviles en los sistemas operativos *iOS* y *Android*. También es usado en navegadores web de uso mayoritario, *Google Chrome* y *Firefox*.

**PostgreSQL**<sup>4</sup> *PostgreSQL* posee licencia *BSD*, es un SGBD relacional, orientado a objetos y libre. A diferencia de *MySQL*, es un proyecto de código abierto y es desarrollado por una amplia comunidad de desarrolladores con el soporte de varias organizaciones.

En 1982 se lanza la primera versión usable. Permite una alta concurrencia y da soporte a gran cantidad de tipos de datos así como la implementación de nuevos tipos. *PostgreSQL* es la elección de empresas como *IMDB*, *Skype* y *Verisign* entre otras para dar soporte a su infraestructura.

**MySQL**<sup>5</sup> *MySQL* es un SGBD relacional, multihilo y multiusuario que cuenta con más de 6 millones de instalaciones, fundado por la empresa de nacionalidad sueca *MySQL AB*. Esta empresa fue absorbida en 2008 por *Sun Microsystems* y esta a su vez en el año 2009 por *Oracle Corporation*, empresa que se encarga en la actualidad de su desarrollo.

Pese a no ser una herramienta Open Source, ya que la propiedad de su código pertenece en su mayoría a una empresa privada, se distribuye bajo licencia *GNU GPL* que permite un uso gratuito del sistema para cualquier fin compatible con dicha licencia. Es compatible con multitud de lenguajes de programación y es la elección mayoritaria, siendo elegida por aplicaciones web muy populares como *Joomla* o *Drupal*. Su popularidad como aplicación web está muy ligada a *PHP*, ya que a menudo aparecen combinadas.

---

<sup>2</sup>JEE: <http://www.oracle.com/technetwork/java/javae>

<sup>3</sup>SQLite: <http://www.sqlite.org/>

<sup>4</sup>PostgreSQL: <http://www.postgresql.org/>

<sup>5</sup>MySQL: <http://www.mysql.com/>

Dispone de diferentes motores de almacenamiento y entre ellos dan soporte a la integridad referencial que es una característica interesante y muy valorable a la hora de implementar un modelo de datos.

## Lenguajes de programación

Con el fin de elegir los lenguajes de programación convenientes, realizaremos una relación de los lenguajes mas populares y los describiremos brevemente:

**HTML**<sup>6</sup> es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a “HyperText Markup Language”, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

*HTML* se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como fotografías, animaciones, etc...) por medio de etiquetas para los navegadores.

**ASP**<sup>7</sup> es el lenguaje del lado del servidor creado por *Microsoft*. El tipo de servidores que emplean este lenguaje son aquellos que funcionan con sistema operativo de la familia de *Windows NT*.

Para escribir páginas *ASP* utilizamos un lenguaje de scripts, que se colocan en la misma página web junto con el código *HTML*. Comúnmente este lenguaje de scripts es *Visual Basic Script*, que deriva del conocido *Visual Basic*, aunque también se pueden escribir los scripts *ASP* en otro lenguaje: *JScript*, que deriva a su vez del conocido *Javascript*.

Debemos mencionar que este lenguaje se ha estancado en su crecimiento y, ha evolucionado a *.NET*.

**PHP**<sup>8</sup> es el lenguaje de lado servidor más extendido en la web. Nacido en 1994, se trata de un lenguaje de creación relativamente reciente, aunque con la rapidez con la que evoluciona Internet parezca que ha existido toda la vida. Es un lenguaje que ha tenido una gran aceptación en la comunidad de desarrolladores, debido a la potencia y simplicidad que lo caracterizan, así como al soporte generalizado en la mayoría de los servidores de hosting.

*PHP* nos permite embeber su pequeños fragmentos de código dentro de las páginas *HTML* y realizar determinadas acciones de una forma fácil y eficaz y, además, ofrece un sinnúmero de funciones para la explotación de bases de datos de una manera llana y sin complicaciones.

**Python**<sup>9</sup> es un lenguaje de programación interpretado con una filosofía que se basa en sintaxis muy limpia y favorecer la legibilidad del código. Se trata de un lenguaje multiplataforma que trabaja en *Windows*, *Linux/Unix* y *Mac OS X*, y ha sido portado a máquinas virtuales *Java* y *.NET*. Por otra parte, posee una licencia de código abierto y es libre incluso para propósitos comerciales.

---

<sup>6</sup>HTML: <http://www.w3.org/html>

<sup>7</sup>ASP: <http://www.asp.net/>

<sup>8</sup>PHP: <http://php.net/>

<sup>9</sup>Python: <http://www.python.org>

Es un lenguaje multiparadigma que permite al programador adoptar su propio estilo, ya sea orientado a objetos, imperativo o funcional.

*Python* está mantenido por *Python Software Foundation*, que posee los derechos de propiedad intelectual de este y de otros proyectos de su comunidad.

**Java**<sup>10</sup> fue desarrollado por *Sun Microsystems*, la cual pertenece ahora a *Oracle*, a principio de los años 90. Se trata de un lenguaje de propósito general, concurrente, orientado a objetos y basado en clases. Se diseñó para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, es decir, el código solo ha de ser compilado una vez.

Actualmente, es uno de los lenguaje mas populares en el desarrollo de aplicación cliente-servidor web.

**Django**<sup>11</sup> es un framework basado en *Python*, con licencia de código abierto BSD, que se basa en el paradigma “Model Template View”. *Django* hace hincapié en el desarrollo rápido y limpio, el re-uso, la conectividad, la extensibilidad y el principio de “DRY”, que procede del inglés “Don’t Repeat Yourself” y que significa “No te repitas”.

Respecto al SGBD, se recomienda el uso de *PostgreSQL*, aunque también soporta *MySQL* o *SQLite 3*. Se prevé una pronta adaptación *Microsoft SQL Server*. Debemos mencionar, que dispone de una API para la abstracción de la base de datos.

## Conclusión

En primer lugar, se desarrollará el sistema sin basarnos en ningún software base. Debido a que en el mercado no existe ningún programa que se pueda extender para conseguir el propósito deseado. Bien por no ajustarse a las necesidades, o bien, por no disponer de licencia de código abierto.

Para la capa de presentación utilizaremos *HTML*. Interesa basarse en la versión mas moderna, *HTML5*, ya que esta ofrece nuevas etiquetas que resultan muy útil para nuestra aplicación. Concretamente la etiqueta “canvas”, que ofrece un lienzo, que representará el campograma del sistema. Esta etiqueta permite al usuario interactuar a través del lenguaje de script *JavaScript*. Por tanto, será necesario hacer uso de éste.

Como lenguaje del lado del servidor se utilizará *PHP*, es decir, será usado para desarrollar la capa de servicios y de datos apoyándonos en el framework *CodeIgniter*, puesto que ya es conocido por el desarrollador. De este modo, no retrasaremos el comienzo de la implementación, aprendiendo a manejar nuevas tecnologías.

Por ser el mas extendido y por ofrecer mayor velocidad en web para *PHP*, elegiremos el SGBD *MySQL*. Además, de que ofrece una completísima herramienta de administración, *phpMyAdmin*.

Por otra parte, deseamos las plataformas tecnológicas pues no son las mas apropiadas para las tecnologías elegidas para el desarrollo del sistema.

---

<sup>10</sup>Java: <https://www.java.com/es>

<sup>11</sup>Django: <https://www.djangoproject.com/>





## Capítulo 4

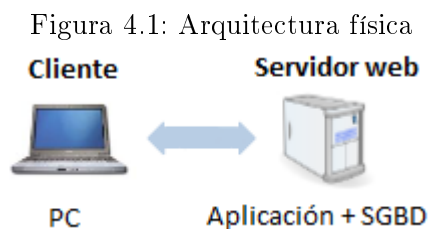
# Diseño del sistema

### 4.1. Diseño de la arquitectura

En esta sección se define la arquitectura general del sistema de información, especificando las distintas particiones físicas del mismo, la descomposición lógica en subsistemas de diseño y la ubicación de cada subsistema en cada partición, así como la especificación detallada de la infraestructura tecnológica necesaria para dar soporte al sistema de información.

#### 4.1.1. Arquitectura física

La arquitectura física del sistema se compone de dos niveles, Cliente y Servidor. En la figura 4.1 se puede apreciar un sencillo esquema.



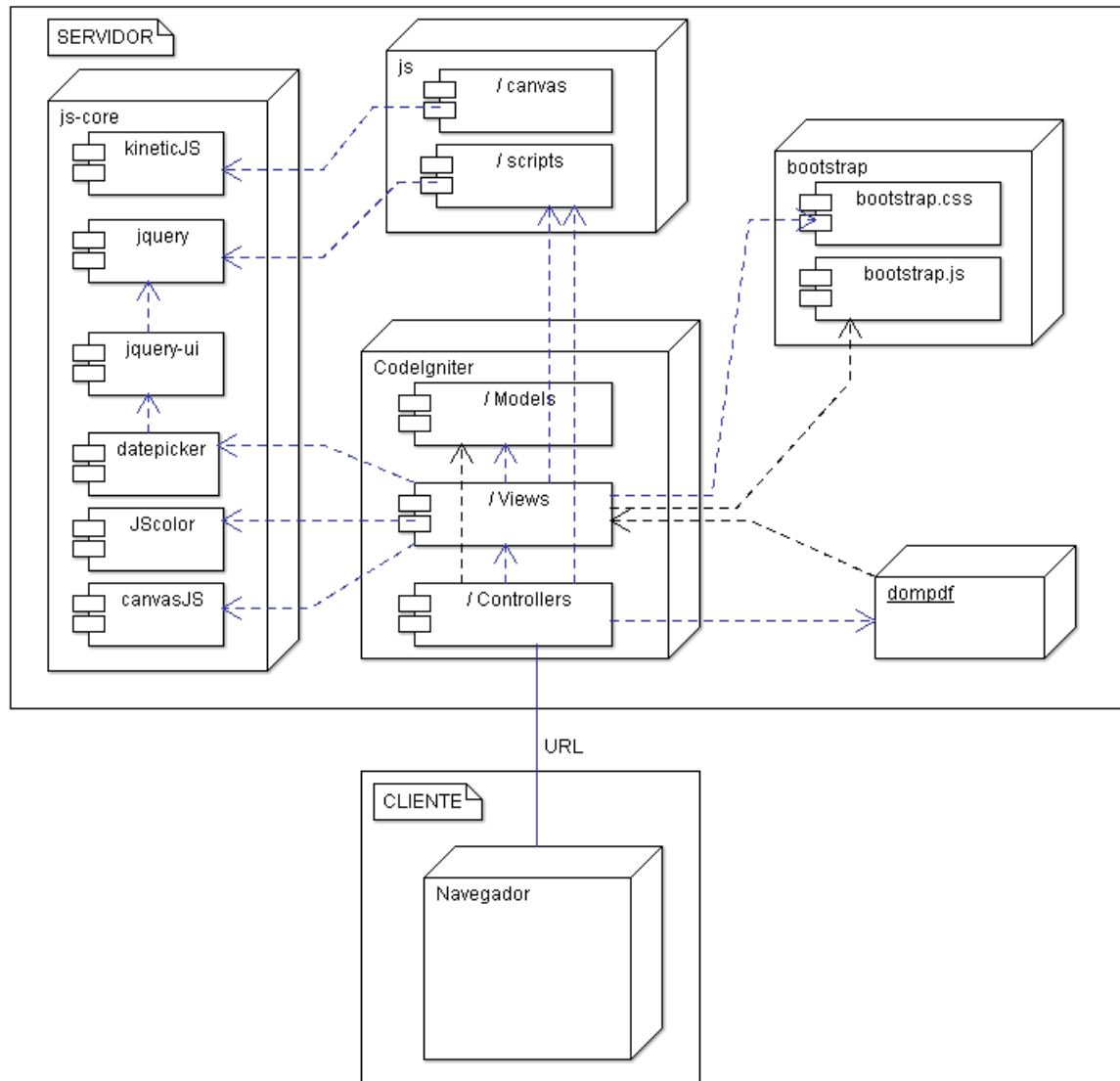
El nivel de clientes lo conforman todos aquellos usuarios que desde sus PCs accedan a la aplicación.

Los componentes del lado del servidor serán el propio servidor web donde residirá la aplicación y el sistema de gestión de base de datos. Durante el desarrollo del sistema hemos utilizado “Apache” con el módulo de *PHP*, servidor web *HTTP* de código abierto y multiplataforma. Y, por otro lado, el sistema de gestión de bases de datos relacional, multihilo y multiusuario, *MySQL*.

#### 4.1.2. Arquitectura lógica

La arquitectura lógica del sistema está formada por los elementos software que componen la aplicación. Esta se puede observar en el diagrama de componentes de la figura 4.2.

Figura 4.2: Arquitectura lógica



#### CLIENTE

**Navegador:** Será el único elemento software necesario en el lado del cliente. Por el hecho de ofrecer calidad, trataremos de desarrollar una aplicación compatible con los mas populares del mercado.

#### SERVIDOR

## Frameworks:

**PHP** Para facilitar la programación *PHP*, se utilizará *CodeIgniter* en la versión 2.1.3, que proporciona su propia estructura basándose en la filosofía Modelo-Vista-Controlador. Dentro de su estructura de ficheros tenemos las carpetas “Models”, “Views” y “Controllers”, donde se ubicarán los ficheros necesarios para desarrollar el sistema.

**CSS** Con el fin de obtener un diseño actual de forma cómoda, se hará uso de *Bootstrap*, un framework que ofrece interfaces web *CSS* y *JavaScript* que adaptan la interfaz al dispositivo para que se visualice adecuadamente.

## Librerías:

**PHP** Para generar ficheros “.pdf”, a partir de código *HTML* o, en este caso, a partir de vistas de *CodeIgniter*, se utilizará la biblioteca *Dompdf* escrita en *PHP*. Es una herramienta ligera que convierte de forma sencilla una página *HTML* en un archivo *PDF*.

**JavaScript** Dentro de la carpeta “js-core” se incluirán las bibliotecas de *JavaScript* que se requieren para el proyecto. Estas librerías serán enumeradas a continuación:

- *Jquery*: Es una biblioteca *JavaScript* que tiene como objetivo simplificar la forma de interactuar con los elementos *HTML*, manipular el árbol del *DOM*, manejar eventos, agregar interacción con *AJAX*, etc...

- *Jquery UI*: Se trata de una biblioteca para *Jquery* que incorpora plugins, widgets y efectos visuales para la creación de aplicaciones web.

- *Datepicker*: Componente de *Jquery UI* que permite tanto mostrar un calendario como seleccionar una fecha de manera visual. Se hará una modificación para señalar en el calendario aquellos días para los que se hayan planificado sesiones de entrenamiento.

- *KineticJS*: Es una biblioteca que extiende el contexto 2D de la etiqueta “canvas” de *HTML5*, de forma que permite interactuar con ella fácilmente. Será utilizada para todo el desarrollo relacionado con el campograma del sistema.

- *JScolor*: Biblioteca ligera de *JavaScript* que permite implementar un selector de color. En el desarrollo del sistema será utilizado para asignar a un club su color y, así, personalizar los ficheros *PDF* que se generen para los clubes.

- *canvasJS*: Se trata de una biblioteca *JavaScript* y *HTML5* que permite representar diagramas casi de cualquier tipo. En el sistema que se está desarrollando será utilizado para mostrar gráficas estadísticas al administrador.

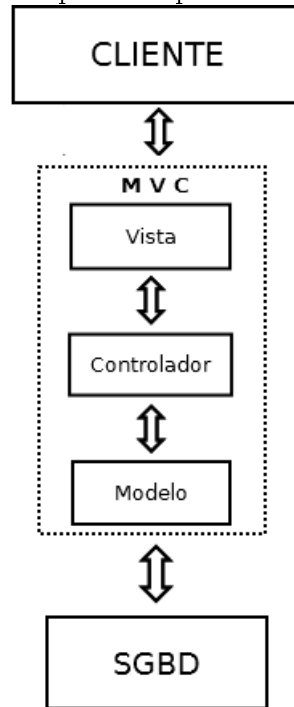
Por otro lado, se incluyen ficheros de *JavaScript* escritos por el desarrollador. Estos serán incluidos en la carpeta “js”, donde se dividirán en dos subcarpetas según el propósito del fichero. La subcarpeta “canvas” contendrá los archivos necesarios para crear, representar el campograma y definir las clases de los elementos que estarán disponibles en el sistema. Por su parte, en la subcarpeta “scripts” se encontrarán los ficheros que permiten la interacción del usuario con el resto de la aplicación.

### 4.1.3. Arquitectura de diseño

La arquitectura de diseño especifica la forma en que los artefactos software de más bajo nivel, interactúan entre sí para lograr el comportamiento deseado en el sistema. El software que se está desarrollando se basa en la filosofía Modelo-Vista-Controlador.

e basan en las ideas de reutilización de código y la separación de conceptos, características que buscan ofrecer calidad y claridad al código y facilitar el mantenimiento.

Figura 4.3: Comparación patrón MVC - Layers



#### Modelo

Representa la interacción con la información de la aplicación con la *SGBD*. De forma que gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso a ella. Puede enviar a la “vista” aquella parte de la información que en cada momento se le solicita para que sea mostrada, o bien, puede ser enviada usando al “controlador” como mediador. Las operaciones de manipulación de información son siempre solicitadas por el “controlador”.

En nuestro sistema las funciones de los ficheros “modelo” contendrán consultas *MySQL*. Nos encontraremos con modelos referentes a los usuarios, campogramas, tareas, estrategias, sesiones y un modelo para el administrador.

#### Vista

Presenta el “modelo” en un formato adecuado para que el usuario pueda interactuar con él. En nuestro caso, los ficheros “vista” contendrán código *HTML* para que puedan ser

interpretados por el navegador del lado del cliente.

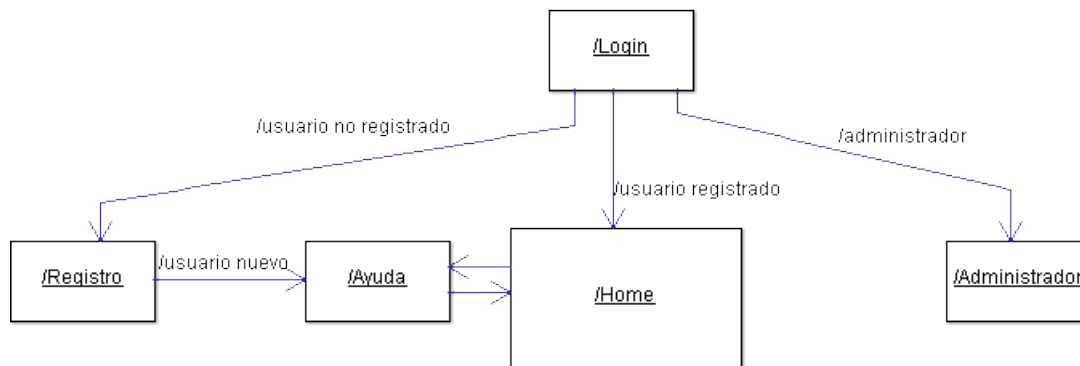
## Controlador

Recibe, interpreta y da respuesta a los eventos enviados tanto por el usuario como por la propia aplicación. Debe interactuar con el “modelo” y con la “vista”. En nuestra aplicación el “controlador” contendrá código *PHP* simplificado con la nomenclatura del framework *CodeIgniter*. Además de esto, el nombre del controlador será parte de la *URL* del sistema.

## 4.2. Diseño de la interfaz de usuario

Para definir el diseño de la interfaz de usuario hemos desarrollado una serie de wireframes<sup>1</sup> de la aplicación, de modo que podamos describir el flujo de navegación de la misma, figura 4.4. En principio, la intención es diseñar una interfaz que conste de una sola página que contemple íntegramente la funcionalidad del sistema. Previamente, el usuario deberá registrarse e identificarse para acceder.

Figura 4.4: Flujo de navegación



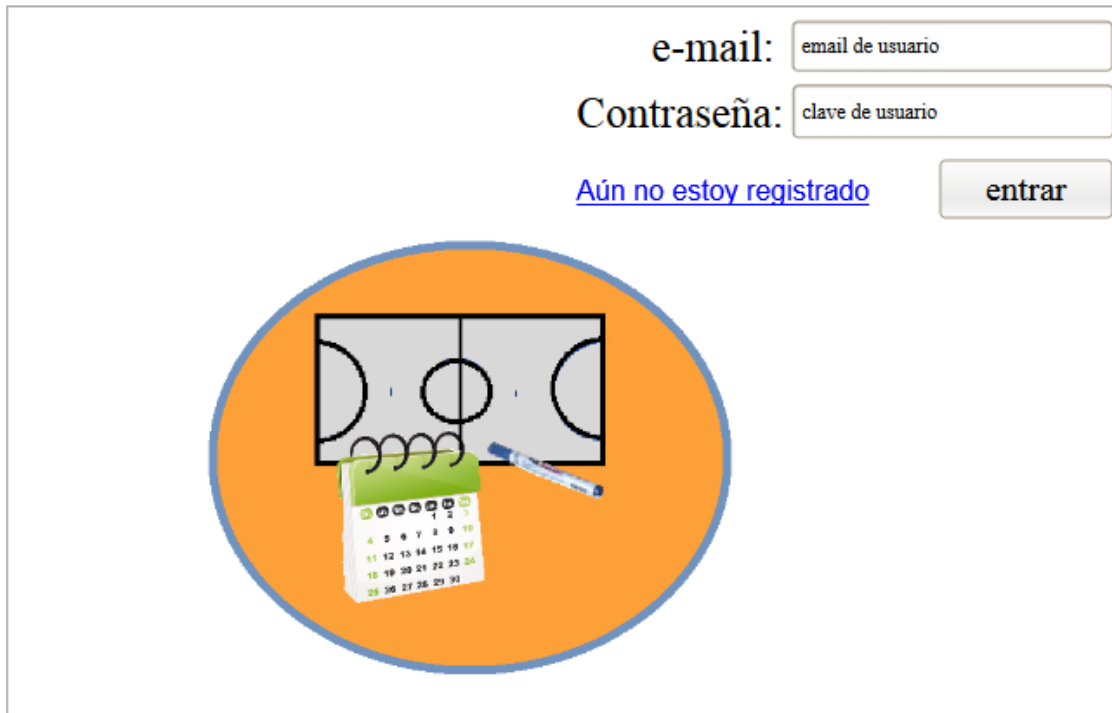
### Login

En primer lugar, el usuario que pretenda acceder a la aplicación deberá identificarse o indicar que aún no está registrado. Esta es la función de “Login” que describimos en la figura 4.5.

---

<sup>1</sup> *Wireframe*: Guía visual que representa el esqueleto o estructura visual de un sitio web.

Figura 4.5: Diseño de interfaz - Login



The login interface is enclosed in a rectangular frame. At the top right, there are two input fields: the first is labeled "e-mail:" and contains the placeholder text "email de usuario"; the second is labeled "Contraseña:" and contains the placeholder text "clave de usuario". Below the password field, there is a blue underlined link that says "Aún no estoy registrado" and a grey button labeled "entrar". In the center of the interface is a large orange circle with a blue border. Inside this circle, there is a grey basketball court diagram, a green spiral-bound calendar, and a blue marker.

## Registro

Los usuarios no registrados deberán hacerlo en la página que describe la figura 4.6 accediendo al enlace que contiene el texto “Aún no estoy registrado”.

Figura 4.6: Diseño de interfaz - Registro



The registration interface is titled "Registro de usuarios" in bold black text. Below the title, it says "Rellene todos los campos a continuación". There are four input fields: the first is labeled "e-mail:" with placeholder text "email de usuario"; the second is labeled "Contraseña:" with placeholder text "clave de usuario"; the third is labeled "Nombre:" with placeholder text "Nombre y apellidos"; and the fourth is labeled "Nombre deportivo:" with placeholder text "alias". At the bottom right of the form is a grey button labeled "¡Registrarse!".

## Home

Una vez registrado e identificado, el usuario tendrá acceso a la aplicación en la que encontrará la página descrita en la figura 4.7.

Haremos hincapié en describir ciertos aspectos de la interfaz que consta de dos partes, cabecera y cuerpo:

**Cabecera:** Aquí nos encontramos un calendario donde estarán señalados los días para los que se ha registrado una sesión de entrenamiento, cierta información del usuario (nombre, alias y foto) y un botón que mostrará y permitirá editar la información del usuario.

**Cuerpo:** De izquierda a derecha podemos ver, una sección con varias listas que son mostradas según en qué esté trabajando el usuario (tareas, estrategias, clubes y sesiones) y, otra con el corazón de la aplicación que será un campograma para las tareas, estrategias y sesiones y, por otro lado, constará de información relativa a los clubes (foto de plantilla y/o escudo).

Para añadir nuevos objetivos, acciones o clubes se ofrecerán los campos de texto necesarios y un botón al final de la sección de las listas. Y, pinchando en cualquier objeto del listado será visualizado a la derecha.

El campograma consta de un lienzo y una serie de herramientas sobre él (jugadores, porteros, líneas, etc..) y dos botones que indican el formato de la pista a mostrar, etc. Además en esta sección encontraremos una serie de botones:

- **Detalle:** Aportará por medio de una ventana emergente la descripción textual de una tarea, estrategia o un club. Cuando nos encontremos en sesiones, mostrará la información de una tarea entre las pertenecientes a la sesión que se esté visualizando. Del mismo modo que visualizaremos el campograma de una sola de estas.
  - **Tarea:** Nombre, Objetivo, Desarrollo, Organización, Reglas y Periodización.
  - **Estrategia:** Nombre, Acción y Desarrollo.
  - **Club:** Nombre, Ciudad e Historial del club.
  - **Sesión:** Detalle de la tarea actual en el visualizador.
- **Información sesión:** Proporciona en una ventana la información textual de la sesión (Lugar, Fecha, Hora, etc...). Mientras visualicemos tareas o estrategias no será posible pulsarlo.
- **< / >:** En principio, la utilidad de estos botones será navegar entre las tareas de la sesión que se esté visualizando, pues a pesar de que una sesión conste de varias tareas, no visualizaremos más de una simultáneamente. No obstante, es probable que se utilicen estos botones en las otras pestañas de la sección de listas para navegar por ellas. Es decir, para ir a tarea/estrategia/club anterior o posterior de la lista.
- **Guardar/Eliminar:** Guarda/Elimina la tarea, estrategia o club que se esté visualizando. Requerirá confirmación del usuario.

Figura 4.7: Diseño de interfaz - Home

Mayo

L	M	X	J	V	S	D
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Tarea

Estrategia

Club

Sesion

- Objetivo 1

- Tarea 1.1

- Tarea 1.2

- Tarea 1.3

...

- Tarea 1.n

+ Objetivo 2

+ Objetivo 3

...

+ Objetivo m

Nombre Objetivo

Descripción

Guardar

Nombre del entrenador

Nombre deportivo

Ficha

Guardar

Eliminar

Añadir a sesión

Quitar

Guardar sesión

Eliminar sesión

Detalle

Información sesión

Imprimible de sesión

71



- **Añadir/Quitar de sesión:** Añade/quita una tarea de la sesión que se esté visualizando.
- **Guardar/Eliminar sesión:** Guarda/elimina la sesión que esté en pantalla. Requiere confirmación del usuario.
- **Imprimible de sesión:** Es el objetivo prioritario de la aplicación. Genera una vista del imprimible de la sesión con la información y el conjunto de tareas a realizar en ella. Una vez visualizada y confirmado que contiene la información y tareas que desea, el usuario podrá imprimir la sesión.

### 4.3. Diseño de datos

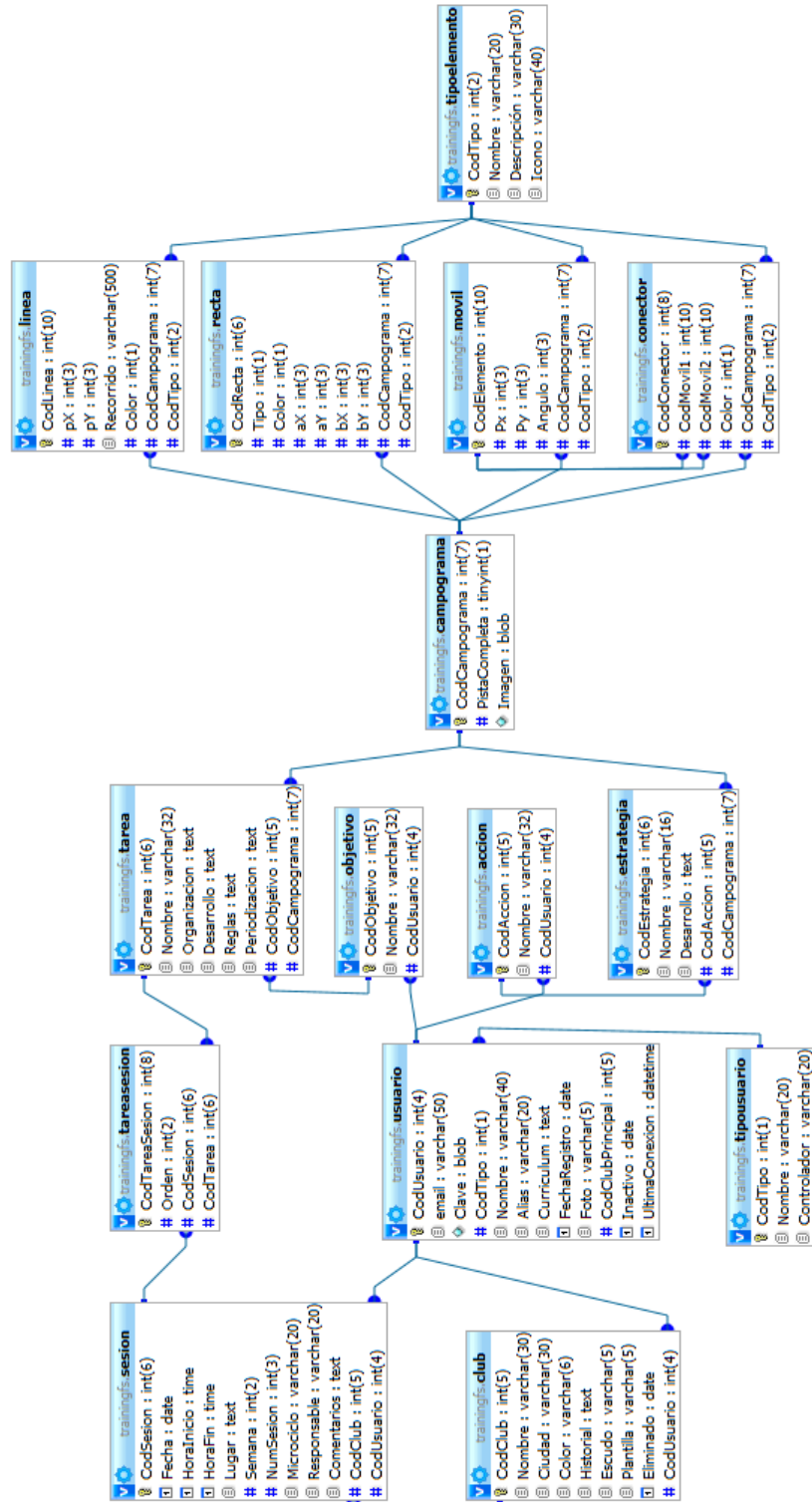
#### Estructura de la base de datos

En la figura 4.8 se puede observar la estructura de la base de datos implementada en MySQL.

#### Descripción de las tablas y las relaciones

- **usuario:** Mantiene la información relativa al usuario de la aplicación.
    - *CodUsuario*: Clave primaria.
    - *email*: Clave candidata, utilizada como identificador de cara al usuario.
    - *Contraseña*: Clave de acceso a la aplicación. Tipo *Blob* permite encriptación.
    - *Nombre*: Nombre del usuario.
    - *Nombre Deportivo*: Alias del usuario.
    - *Curriculum*: Historial deportivo del usuario. Lo registrará él mismo durante el uso de la aplicación.
    - *FechaRegistro*: Fecha en la que usuario es dado de alta en la aplicación.
    - *Foto*: Booleano para saber si el usuario ha subido su foto personal. La aplicación contendrá una carpeta donde almacenará estas fotos. El formato de la ruta será “/FotoEntrenador/CodUsuario.ext”.
    - *CodClubPrincipal*: Lo definirá el usuario para saber en que club está trabajando en el momento.
    - *Inactivo*: Fecha que definirá si el usuario tiene activa su cuenta en la aplicación. Mientras esté activo será *NULL*, y cuando se pretenda dar de baja registraremos la fecha.
    - *CodTipo*: Identifica al tipo de usuario. Clave foránea para la relación con la tabla “tipoUsuario”. [ usuario.CodTipo - tipoUsuario.CodTipo ]
      - **ON DELETE:** No action
- No será posible eliminar un tipo que esté asociado a algún usuario.
- **ON UPDATE:** Cascade

Figura 4.8: Diseño de datos



Aunque no sea usual, si se modifica el valor de la clave primaria de la tabla de tipos de usuario, se actualizará en cascada el campo correspondiente de los registros referenciados de la tabla “usuario”.

- **club:** Gestiona la información de los clubes de los usuarios.
  - *CodClub*: Clave primaria.
  - *Nombre*: Nombre del club.
  - *Ciudad*: Ciudad donde tiene sede el club.
  - *Historial*: Breve descripción del club. Esta información la registrará el usuario.
  - *Escudo*: Booleano que confirma si el usuario ha subido la foto del escudo del club. La ruta tendrá el forma “/FotoEscudo/CodClub.ext”.
  - *Plantilla*: Booleano que confirma si el usuario ha subido la foto de la plantilla del club. La ruta tendrá el forma “/FotoClub/CodClub.ext”.
  - *CodUsuario*: Representa al usuario que ha creado del club. Clave foránea para la relación con la tabla “usuario”. [ club.CodUsuario - usuario.CodUsuario ]
    - **ON DELETE:** Cascade  
Normalmente para dar de baja a un usuario utilizaremos el campo “Inactivo”. Sin embargo, si en alguna situación decidimos eliminar a un usuario del sistema, eliminaremos toda la información que haya registrado.
    - **ON UPDATE:** Cascade  
Del mismo modo, si existe algún motivo por el que necesitemos cambiar el valor del identificador del usuario, mantendremos asociada la información que éste haya registrado actualizando el campo referenciado en las tablas hijas.
- **objetivo:** Registra los objetivos definidos por los usuarios que clasifican las tareas.
  - *CodObjetivo*: Clave primaria.
  - *Nombre*: Nombre que define el objetivo.
  - *Descripción*: Define una breve descripción del objetivo.
  - *CodUsuario*: Representa al usuario que lo ha registrado. Clave foránea para la relación con la tabla “usuario”. [ objetivo.CodUsuario - usuario.CodUsuario ]
    - **ON DELETE:** Cascade  
Normalmente para dar de baja a un usuario utilizaremos el campo “Inactivo”. Sin embargo, si en alguna situación decidimos eliminar a un usuario del sistema, eliminaremos toda la información que haya registrado.
    - **ON UPDATE:** Cascade  
Del mismo modo, si existe algún motivo por el que necesitemos cambiar el valor del identificador del usuario, mantendremos asociada la información que éste haya registrado actualizando el campo referenciado en las tablas hijas.

- **tarea:** Almacena la información relativa a los ejercicios definidos por los usuarios.
  - *CodTarea*: Clave primaria.
  - *Nombre*: Nombre de la tarea.
  - *Organización, Desarrollo, Reglas y Periodización*: Representan la descripción textual de la tarea.
  - *CodObjetivo*: Asocia la tarea con el objetivo al que pertenece. Clave foránea para la relación con la tabla “objetivo” [ tarea.CodObjetivo - objetivo.CodObjetivo ]
    - **ON DELETE**: No action  
El sistema no permitirá eliminar un objetivo si existen registros de la tabla de tareas que referencian a dicho objetivo.
    - **ON UPDATE**: Cascade  
Esta es una situación que debería ser anormal en el uso del sistema. No obstante, si necesitamos cambiar el valor de la clave primaria de algún registro de la tabla “objetivo”, los valores de clave foránea referenciados “tarea.CodObjetivo” serán actualizados en cascada.
  - *CodCampograma*: Vincula la tarea al campograma que supone la descripción gráfica. Clave foránea de la relación con la tabla “campograma”. [ tarea.CodCampograma - campograma.CodCampograma ]
    - **ON DELETE**: No action  
El sistema no permitirá eliminar un campograma si está asociado a una tarea.
    - **ON UPDATE**: Cascade  
Durante el uso de la aplicación no debería darse esta situación, ya que el valor de clave primaria de un campograma debería mantenerse para siempre. Sin embargo, en el caso de que fuese necesario, el valor la clave foránea también será actualizada.
- **sesión:** Gestiona la información de las sesiones creadas por el usuario.
  - *CodSesion*: Clave primaria.
  - *FechaHora, HoraFin, Lugar, Semana, NumSesion, Microciclo, Responsable, Comentarios*: Representan la información de la sesión.
    - *FechaHora*: Refleja el comienzo de la sesión.
    - *HoraFin*: El final de la sesión.
    - *Lugar*: Instalaciones deportivas donde se realizará.
    - *Semana*: El numero de semana de trabajo.
    - *NumSesion*: El numero de sesión de trabajo.
    - *Microciclo*: Descripción del microciclo.
    - *Responsable*: Personal del cuerpo técnico encargado de llevar a cabo la sesión.
    - *Comentarios*: Espacio para comentarios previos o posteriores a la sesión.
  - *CodClub*: Representa al club que realizará la sesión. Clave foránea de la relación con la tabla “club”. [sesion.CodClub - club.CodClub ]

- **ON DELETE:** Set NULL  
Si una sesión está diseñada para un club y el usuario decide eliminarlo, mantendremos la sesión pero consideraremos que no está dedicada a ningún club en especial. Para ellos utilizaremos esta subcláusula, que hace NULL el campo de clave foránea de las tablas hijas que referencian al registro de la tabla padre eliminado.
  - **ON UPDATE:** Cascade  
Actualizar el valor de clave primaria de un club es una situación que debería ser anormal en el uso del sistema. No obstante, si lo necesitáramos, los valores de clave foránea referenciados “sesion.CodClub” serían actualizados en cascada.
- *CodUsuario*: Define al usuario que ha registrado la sesión. Clave foránea para la relación con la tabla “usuario”. [ sesion.CodUsuario - usuario.CodUsuario ]
  - **ON DELETE:** Cascade  
Normalmente para dar de baja a un usuario utilizaremos el campo “Inactivo”. Sin embargo, si en alguna situación decidimos eliminar a un usuario del sistema, eliminaremos toda la información que haya registrado.
  - **ON UPDATE:** Cascade  
Del mismo modo, si existe algún motivo por el que necesitemos cambiar el valor del identificador del usuario, mantendremos asociada la información que éste haya registrado actualizando el campo referenciado en las tablas hijas.
- **tareassesion**: Se trata lo que teóricamente denominaríamos una clase de asociación entre tareas y sesiones. Pues una sesión ha de contener varias tareas, y por su parte, una misma tarea puede ser incluida en varias sesiones. Por tanto contiene las tareas asociadas a las sesiones y el orden que ocupa dentro de ella.
  - *CodTareaSesion*: Clave primaria.
  - *Orden*: Almacena la posición que ocupa la tarea en la sesión.
  - *CodSesion*: Representa la sesión. Clave foránea para la relación con la tabla “sesion” [ tareassesion.CodSesion - sesion.CodSesion ].
    - **ON DELETE:** Cascade  
Si decidimos eliminar una sesión, lógicamente todos los registros de la tabla “tareassesion” que referencien a ésta serán eliminados en cascada.
    - **ON UPDATE:** Cascade  
Si existiera alguna razón por la que necesitáramos cambiar el valor del identificador de la sesión, mantendremos asociados los registros de la tabla “tareassesion” actualizando el valor del campo que la referencia.
  - *CodTarea*: Representa la tarea. Clave foránea de la relación con la tabla “tarea” [tareassesion.CodTarea - tarea.CodTarea ].
    - **ON DELETE:** Cascade  
Se eliminará el registro de tareassesion. Esto significa que el numero de tareas para las sesiones en que esté incluida esa tarea disminuirá.

- **ON UPDATE:** Cascade  
Aunque no sea usual, si modificamos el valor de la clave primaria de la tabla de tareas, actualizaremos en cascada el campo correspondiente de los registros referenciados de la tabla “tareasesion”.
- **accion:** Registra las acciones definidas por los usuarios que clasifican las estrategias.
  - *CodAccion*: Clave primaria.
  - *Nombre*: Nombre que define la acción.
  - *Descripción*: Define una breve descripción de la acción.
  - *CodUsuario*: Representa al usuario que la ha registrado. Clave foránea para la relación con la tabla “usuario”. [ accion.CodUsuario - usuario.CodUsuario ]
    - **ON DELETE:** Cascade  
Normalmente para dar de baja a un usuario utilizaremos el campo “Inactivo”. Sin embargo, si en alguna situación decidimos eliminar a un usuario del sistema, eliminaremos toda la información que haya registrado.
    - **ON UPDATE:** Cascade  
Del mismo modo, si existe algún motivo por el que necesitemos cambiar el valor del identificador del usuario, mantendremos asociada la información que éste haya registrado actualizando el campo referenciado en las tablas hijas.
- **estrategia:** Almacena la información relativa a las estrategias definidas por los usuarios.
  - *CodEstrategia*: Clave primaria.
  - *Nombre*: Nombre de la estrategia.
  - *Desarrollo*: Representa la descripción textual de la estrategia.
  - *CodAccion*: Clave foránea. Asocia la estrategia con la acción a la que pertenece.
    - **ON DELETE:** No action  
El sistema no permitirá eliminar una acción si existen registros de la tabla de estrategias que referencian a dicha acción.
    - **ON UPDATE:** Cascade  
Esta es una situación que debería ser anormal en el uso del sistema. No obstante, si necesitamos cambiar el valor de la clave primaria de algún registro de la tabla “accion”, los valores de clave foránea referenciados “estrategia.CodAccion” serán actualizados en cascada
  - *CodCampograma*: Vincula la estrategia al campograma que supone la descripción gráfica. Clave foránea para la relación con la tabla “campograma” [ estrategia.CodCampograma - campograma.CodCampograma ]
    - **ON DELETE:** No action  
El sistema no permitirá eliminar un campograma si está asociado a una estrategia.

- **ON UPDATE:** Cascade  
Durante el uso de la aplicación no debería darse esta situación, ya que el valor de clave primaria de un campograma debería mantenerse para siempre. Sin embargo, en el caso de que fuese necesario, el valor la clave foránea también será actualizada.
- **campograma:** Representa los campogramas que estarán asociados a las tareas y las estrategias.
  - *CodCampograma*: Clave primaria.
  - *PistaCompleta*: Booleano que representa el formato de la pista que se mostrará (Pista completa o media pista)
- **tipoelemento:** Describe los tipos de elementos (herramientas gráficas) que podrán representarse en los campogramas. Existirá un registro de esta tabla para cada tipo de elemento.
  - *CodTipo*: Clave primaria.
  - *Nombre*: Nombre de la herramienta.
  - *Descripción*: Breve descripción del tipo o del uso de éste.
  - *Icono*: Imagen que representará al tipo en el campograma.
- **movil:** Almacena la información de los elementos móviles y girables. Son elementos móviles los jugadores, porteros, balones y conos. Y, por otra parte, los móviles girables son las porterías y los bancos.
  - *CodElemento*: Clave primaria.
  - *X*: Tipo *int* que representa la posición horizontal que ocupa el elemento en el campograma.
  - *Y*: Tipo *int* que representa la posición vertical que ocupa el elemento en el campograma.
  - *Angulo*: Representa la rotación del elemento. Para los elementos no girables este campo será *NULL*.
  - *CodTipo*: Indica el tipo de elemento del que se trata. Clave foránea para la relación con la tabla “tipoElemento” [ movil.CodTipo - tipoElemento.CodTipo].
    - **ON DELETE:** No action  
El sistema no debe permitir eliminar un tipo de elementos mientras exista registrado alguno de dicho tipo. Por tanto, utilizamos esta subcláusula con la que no es posible eliminar un registro de “tipoelemento”, mientras existan valores de clave foránea que lo referencien en las tablas hijas.
    - **ON UPDATE:** Cascade  
En el futuro puede considerarse oportuno añadir tipos de elementos y, por alguna razón modificar el valor de clave primaria de alguno de los tipos ya registrados. De modo que, los elementos que referencien en las tablas hijas actualizarán el valor de su clave foránea.

- *CodCampograma*: Indica al campograma al que está asociado. Clave foránea de la relación con la tabla “campograma” [ movil.CodCampograma - campograma.CodCampograma].
  - **ON DELETE**: Cascade  
Eliminar un campograma supondrá eliminar todos los elementos incluidos en él.
  - **ON UPDATE**: Cascade  
Esta es una situación que no está prevista durante el uso de la aplicación. No obstante, consideramos que actualizar el valor de CodCampograma de un campograma supondrá actualizar el valor de sus elementos para que se mantengan incluidos.
- **conector**: Almacena la información de los elementos conectores definidos por el usuario.
  - *CodConector*: Clave primaria.
  - *CodMovil1 y CodMovil2*: Indican los móviles que une el conector. Claves foráneas para la relación con la tabla “movil” [conector.CodMovil - movil.CodMovil].
    - **ON DELETE**: Cascade  
En el momento que se elimine uno de los móviles que une el elemento conector, éste debe ser eliminado, ya que un conector con sólo un móvil no tiene sentido.
    - **ON UPDATE**: Cascade  
Si por alguna razón, no conocida en este momento, algún móvil varía su valor de clave primaria, los valores de clave foránea de los registros referenciados en la tabla hija “conector” serán actualizados.
  - *CodTipo*: Indica el tipo de elemento del que se trata. Clave foránea para la relación con la tabla “tipoElemento” [ conector.CodTipo - tipoElemento.CodTipo].
    - **ON DELETE**: No action  
El sistema no debe permitir eliminar un tipo de elementos mientras exista registrado alguno de dicho tipo. Por tanto, utilizamos esta subcláusula con la que no es posible eliminar un registro de “tipoelemento”, mientras existan valores de clave foránea que lo referencien en las tablas hijas.
    - **ON UPDATE**: Cascade  
En el futuro puede considerarse oportuno añadir tipos de elementos y, por alguna razón modificar el valor de clave primaria de alguno de los tipos ya registrados. De modo que, los elementos que referencien en las tablas hijas actualizarán el valor de su clave foránea.
  - *CodCampograma*: Indica al campograma al que está asociado. Clave foránea de la relación con la tabla “campograma” [ conector.CodCampograma - campograma.CodCampograma].
    - **ON DELETE**: Cascade  
Eliminar un campograma supondrá eliminar todos los elementos incluidos en él.



- **ON UPDATE:** Cascade  
Esta es una situación que no está prevista durante el uso de la aplicación. No obstante, consideramos que actualizar el valor de CodCampograma de un campograma supondrá actualizar el valor de sus elementos para que se mantengan incluidos.
- **linea:** Registra la información de las líneas que trazaran los usuarios.
  - *CodLinea*: Clave primaria.
  - *Recorrido*: Representa el trazo de la línea. Utilizamos el tipo *linestring* que es una curva con interpolación de puntos.
  - *CodTipo*: Indica el tipo de elemento del que se trata. Clave foránea para la relación con la tabla “tipoElemento” [ linea.CodTipo - tipoElemento.CodTipo].
    - **ON DELETE:** No action  
El sistema no debe permitir eliminar un tipo de elementos mientras exista registrado alguno de dicho tipo. Por tanto, utilizamos esta subcláusula con la que no es posible eliminar un registro de “tipoelemento”, mientras existan valores de clave foránea que lo referencien en las tablas hijas.
    - **ON UPDATE:** Cascade  
En el futuro puede considerarse oportuno añadir tipos de elementos y, por alguna razón modificar el valor de clave primaria de alguno de los tipos ya registrados. De modo que, los elementos que referencien en las tablas hijas actualizarán el valor de su clave foránea.
  - *CodCampograma*: Indica al campograma a la que está asociada. Clave foránea de la relación con la tabla “campograma” [ linea.CodCampograma - campograma.CodCampograma].
    - **ON DELETE:** Cascade  
Eliminar un campograma supondrá eliminar todos los elementos incluidos en él.
    - **ON UPDATE:** Cascade  
Esta es una situación que no está prevista durante el uso de la aplicación. No obstante, consideramos que actualizar el valor de CodCampograma de un campograma supondrá actualizar el valor de sus elementos para que se mantengan incluidos.

## 4.4. Diseño de componentes

### Descripción de las clases

Los componentes se han estructurado bajo la filosofía Modelo-Vista-Controlador. Por tanto, se encontrará un controlador y un modelo para cada uno de los siguientes paquetes funcionales. Por otra parte, aunque se organizarán las vistas bajo los mismos paquetes, cada uno de estos estará compuesto por una carpeta que contendrá con un conjunto de ficheros vista y que se subdividirá si es necesario.

**Home** Contiene los ficheros necesarios para que el usuario trabaje con el sistema ofrecido de inicio tras identificarse. Es decir, contiene el menú y las vistas principales por defecto para los usuarios.

**Usuario** Conjunto de ficheros necesarios para registrar al usuario, identificarlo y mostrar su información. Las vistas de este paquete estarán subdividas en *Login* y *Perfil*.

**Tarea** Permite crear, editar, eliminar, listar y visualizar las tareas por parte del usuario.

**Estrategia** Contiene los ficheros para ofrecer las funcionalidades de las estrategias.

**Club** Comprende los archivos necesarios para gestionar los clubes.

**Sesion** Ofrece las funciones de crear, editar, eliminar, ver, listar en el calendario e imprimir las sesiones de entrenamiento.

**Administrador** Permite al administrador ver estadísticas de uso de la aplicación y modificar las cuentas de los usuarios registrados.

**Ayuda** Ofrece información útil al usuario para la comprensión del sistema.

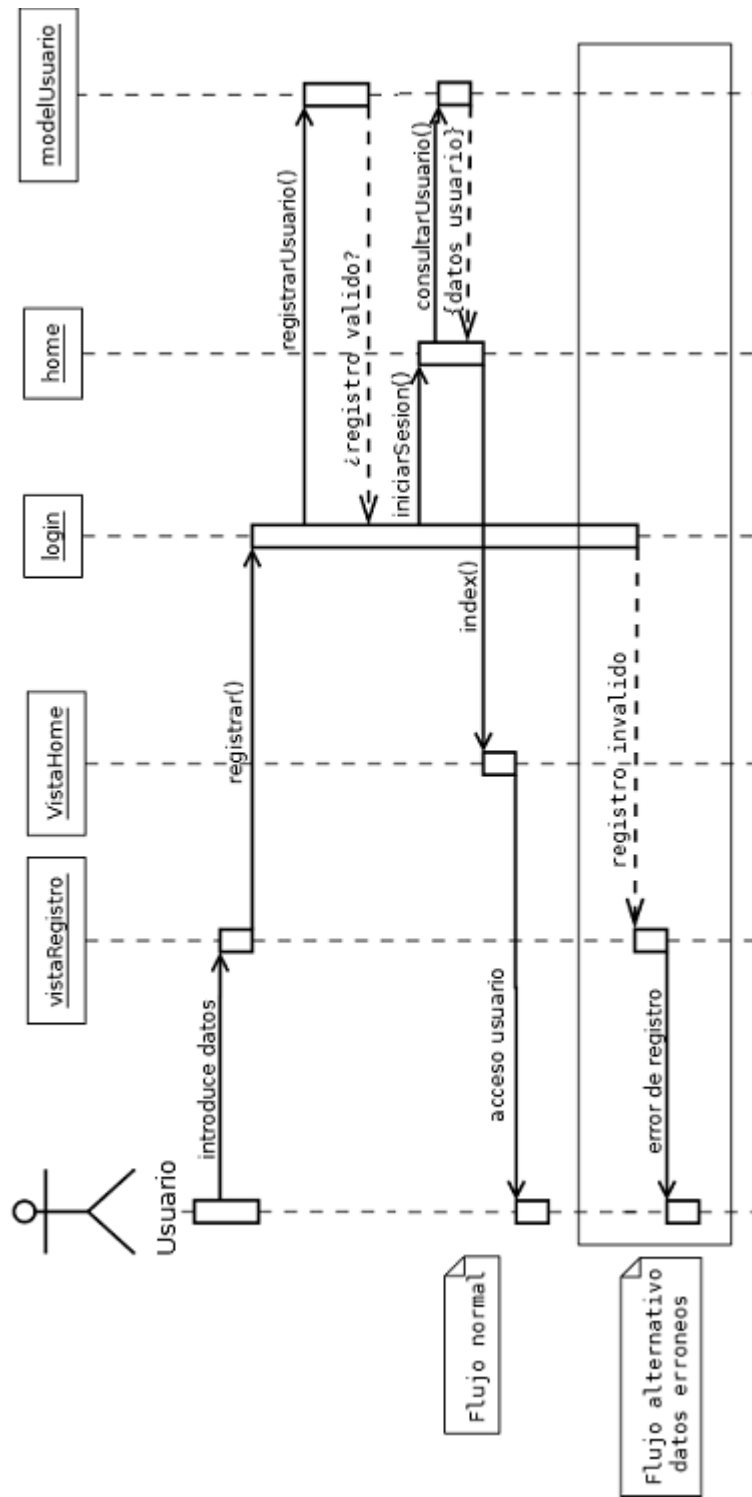
## Diagramas de secuencia

Una vez vistos las clases de programación, se mostrarán los diagramas de secuencia de las mismas. De esta forma, se observará la relación que existe entre los distintos módulos de cada paquete funcional.

### Usuario

En primer lugar, se observarán los diagramas de secuencia relativos a las gestión de usuarios. En la figura 4.9 se muestra el proceso de registro de usuarios y en la figura 4.10 la identificación necesaria para acceder al sistema. Por otra parte, el proceso de recuperación de cuenta de usuario seguirá una secuencia similar al registro con la vista correspondiente. Por su parte, la figura 4.11 muestra la interacción necesaria para editar la información del perfil de usuario.

Figura 4.9: Registrar usuario



*registro Valido* contiene información sobre la validez de los datos introducidos por el usuario. De tal modo que cargará la vista correspondiente en función de esta información. Si se introducen datos correctos, el sistema cargará la vista *Home*. Pero si se introducen datos erróneos, el sistema cargará de nuevo la vista *Registro*.

```

sequenceDiagram
    actor Usuario
    participant VistaLogin
    participant VistaHome
    participant VistaAdmin
    participant login
    participant home
    participant admin
    participant modelUsuario
    participant modelAdmin

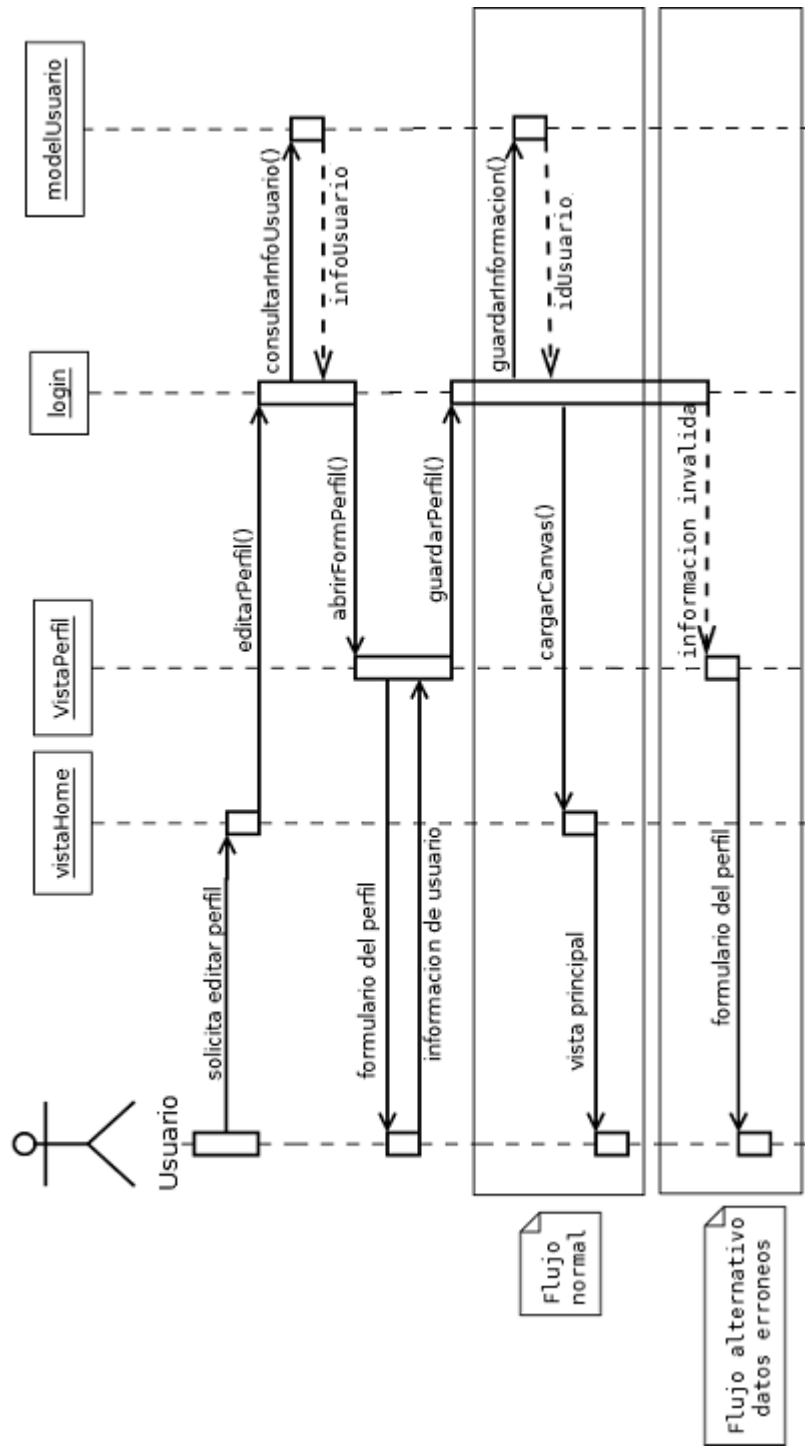
    Usuario->>VistaLogin: introduce datos
    activate VistaLogin
    VistaLogin->>VistaHome: identificar()
    deactivate VistaLogin
    activate VistaHome
    VistaHome->>login: acceso usuario
    deactivate VistaHome
    activate login
    login->>modelUsuario: consulta usuario()
    deactivate login
    activate modelUsuario
    modelUsuario-->>login: datos usuario
    deactivate modelUsuario
    login->>home: iniciar sesion()
    deactivate login
    activate home
    home-->>login: ¿login valido?
    deactivate home
    login->>admin: acceder()
    deactivate login
    activate admin
    admin->>modelAdmin: consultar usuarios()
    deactivate admin
    activate modelAdmin
    modelAdmin-->>admin: datos usuarios
    deactivate modelAdmin
    deactivate admin

    VistaLogin->>VistaAdmin: acceso administrador
    deactivate VistaLogin
    activate VistaAdmin
    VistaAdmin->>admin: index()
    deactivate VistaAdmin
    activate admin
    deactivate admin

    VistaLogin->>VistaAdmin: login invalido
    deactivate VistaLogin
    activate VistaAdmin
    VistaAdmin->>VistaAdmin: acceso denegado
    deactivate VistaAdmin
    deactivate VistaAdmin
  
```

*vista Login.*

Figura 4.11: Editar perfil de usuario



*InformacionInvalida* contiene información sobre la validación del formulario. De esta forma avisa al usuario de los posibles errores cometidos al introducir la información.

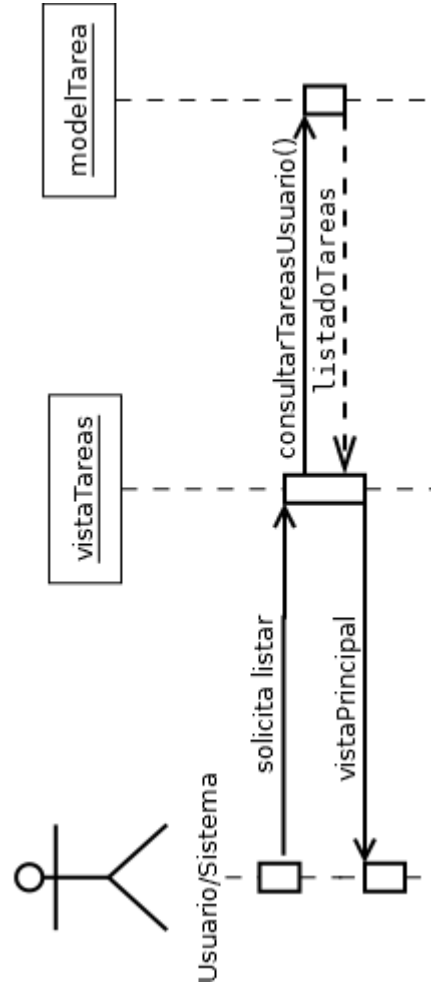
Si la validación es correcta, actualiza los datos del servidor a través del modelo y carga de nuevo la vista principal del sistema. La función *cargarCanvas()* se encargará de volver a mostrar el campograma en la sección central del sistema, ya que es ahí donde se mostrará el formulario. Este procedimiento se sucederá en la mayoría de las funcionalidades.

## **Tarea**

La siguiente relación de diagramas que pueden apreciarse corresponden al paquete funcional de las tareas.

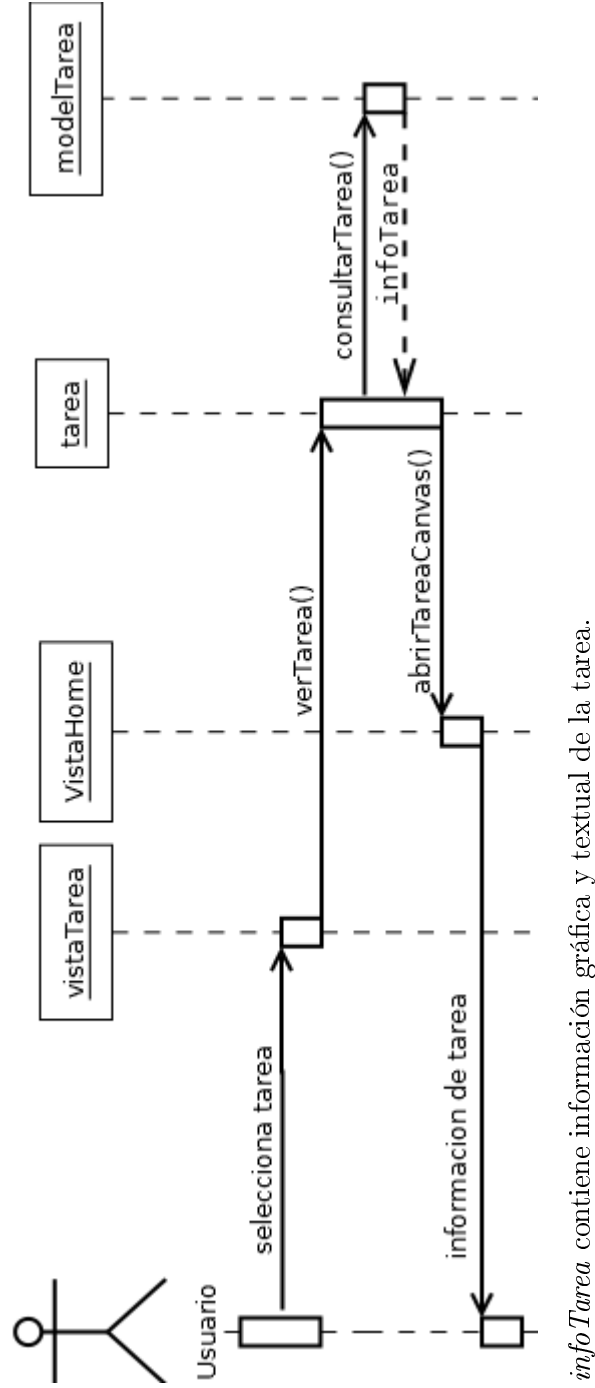
En la figura 4.13 se puede observar la secuencia de procesamiento para ver una tarea. La figura 4.14 representa la interacción necesaria entre clases para crear una tarea nueva. Por su parte, para editar una tarea primero habrá que visualizarla y, posteriormente, proceder del mismo modo que para crearla. Por otro lado, el diagrama de secuencia para eliminar una tarea puede verse en la figura 4.15.

Figura 4.12: Listar tareas



*listadoTareas* contiene una lista de tareas del usuario. Esta funcionalidad no requiere que el controlador procese ninguna información, así que la vista solicita el listado directamente al modelo.

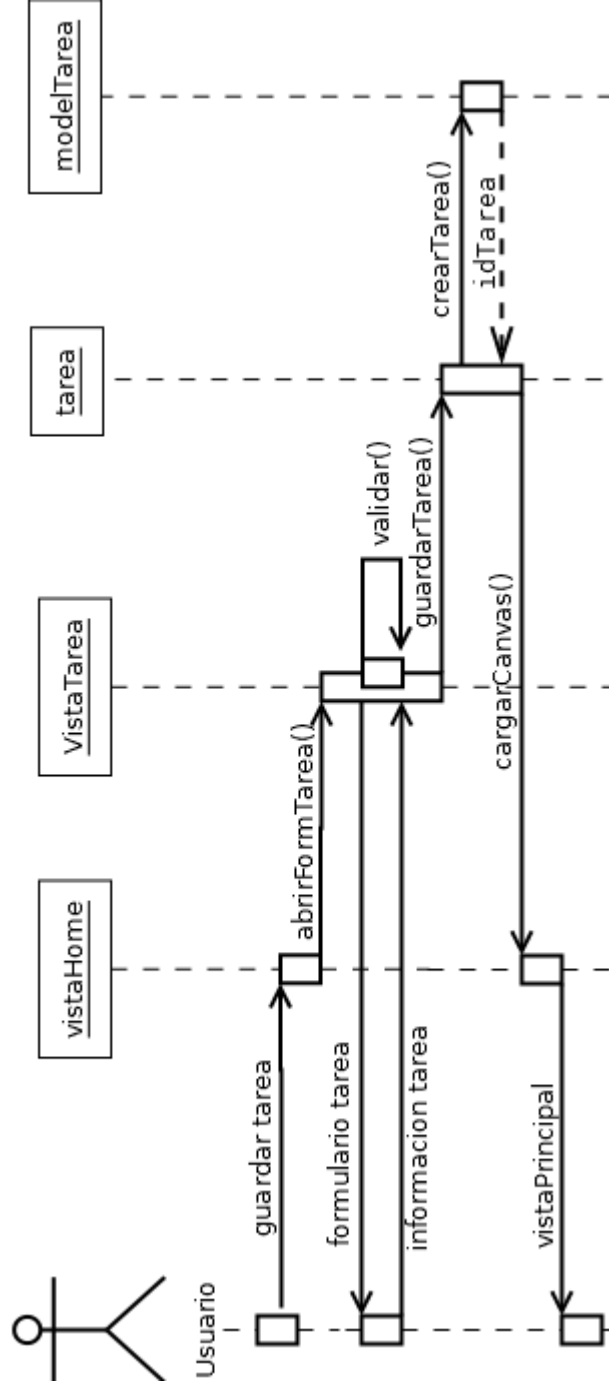
Figura 4.13: Visualizar tarea



*infoTarea* contiene información gráfica y textual de la tarea.

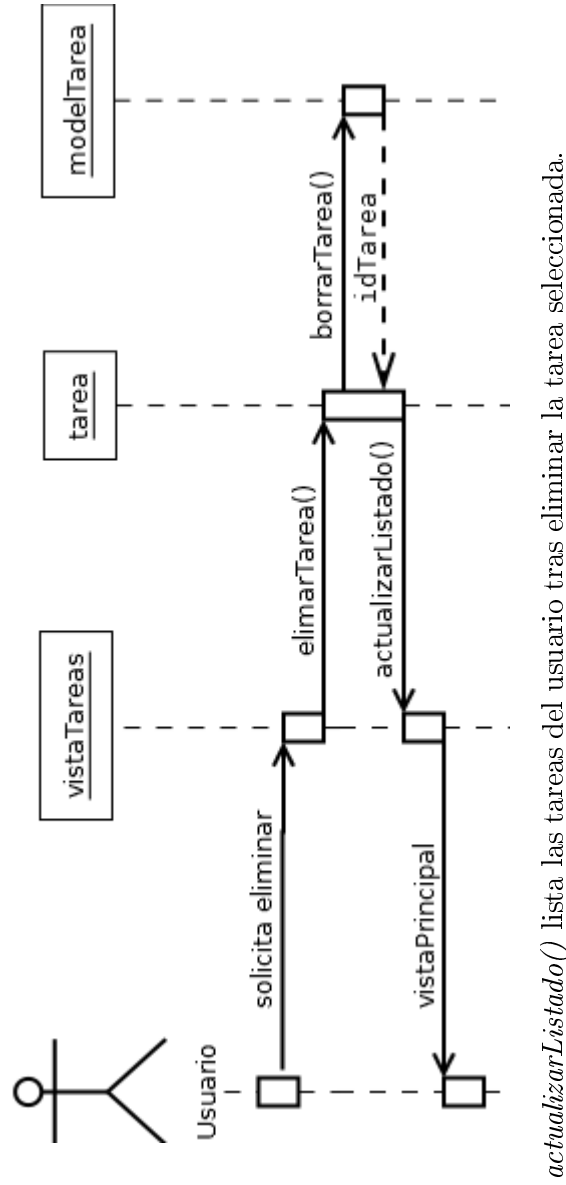


Figura 4.14: Registrar tarea



*validar()* es una función que comprueba si la información del formulario es correcta. Si esta información no es válida cargará de nuevo la vista notificando al usuario el error. En esta ocasión el formulario solo necesitará obligatoriamente el nombre de la tarea y, por esto, no lo hemos considerado un flujo alternativo.

Figura 4.15: Eliminar tarea

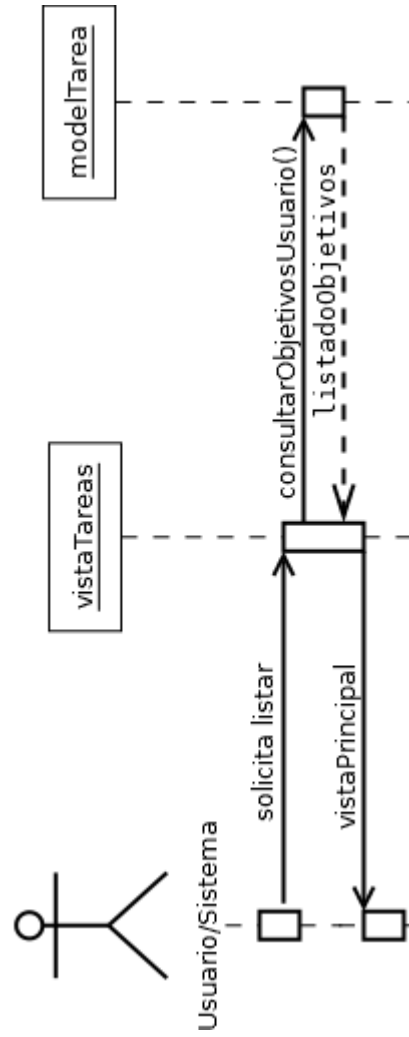


*actualizarListado()* lista las tareas del usuario tras eliminar la tarea seleccionada.

Ademas de estas, la gestión de tareas incluye funciones relativas a los objetivos. En la figura Listar objetivos se puede observar la interacción de las clases para listar los objetivos y para registrar un nuevo objetivo se sigue la secuencia que muestra la figura Registrar objetivo.

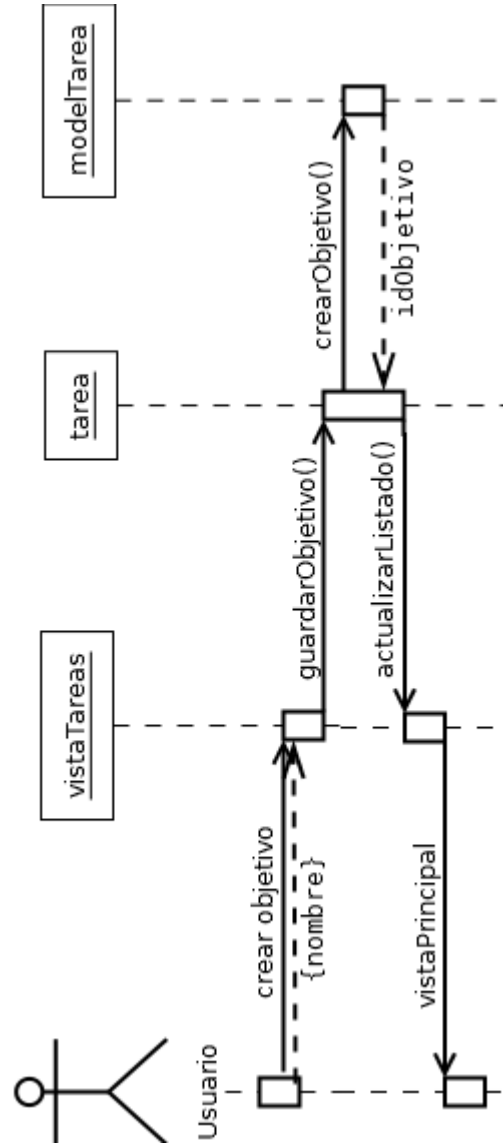
Por otro lado, editar un objetivo implica una secuencia similar al registro, pues solo es necesario introducir el nuevo nombre. Mientras que por parte de la eliminación de un objetivo supone la misma secuencia de procesamiento que eliminar una tarea. De modo que no incluiremos los diagramas correspondientes a estas funciones.

Figura 4.16: Listar objetivos



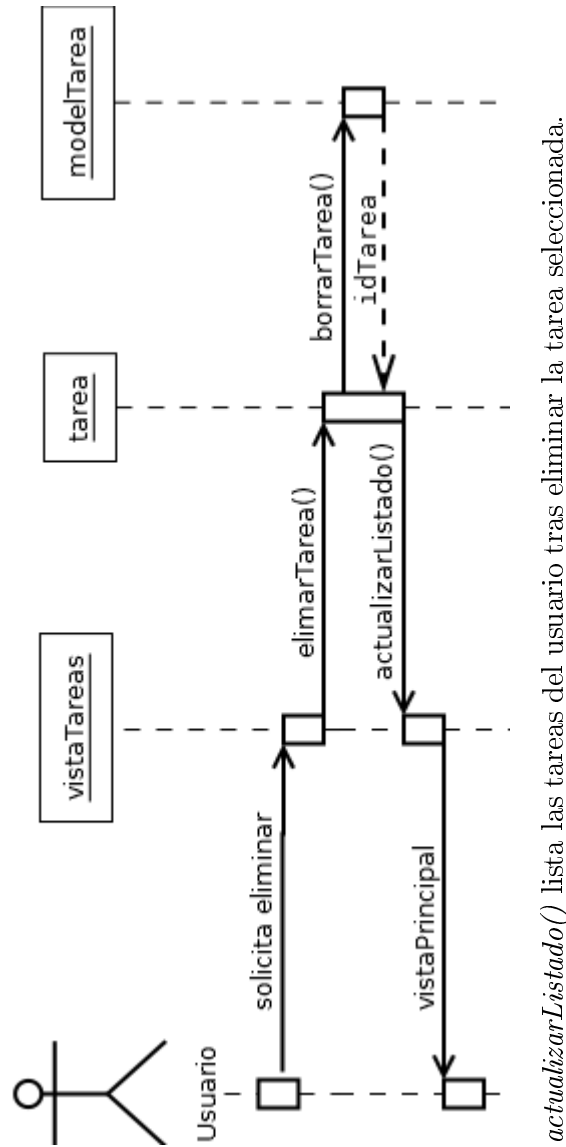
*listadoTareas* contiene una lista de tareas del usuario. Esta funcionalidad no requiere que el controlador procese ninguna información, así que la vista solicita el listado directamente al modelo.

Figura 4.17: Registrar objetivo



Para crear un objetivo tan solo será necesario introducir el nombre que le será asignado.

Figura 4.18: Eliminar objetivo



*actualizarListado()* lista las tareas del usuario tras eliminar la tarea seleccionada.

## **Estrategia**

Las funcionalidades de este paquete tienen su homólogo en la gestión de tareas. Las funciones propias de estrategias se corresponden con las de tareas y las de acciones con las de objetivos.

Por tanto, no se desarrollarán los diagramas de secuencia de la gestión de estrategias. Véase la sección Tarea detallada anteriormente.

## **Club**

Del mismo modo que las estrategias, la funcionalidad de los clubes tienen secuencias similares a las descritas para la gestión de tareas. Véase la sección Tarea detallada anteriormente.

## **Sesion**

A continuación se muestran los diagramas de secuencia para las funcionalidades de las sesiones. Tan solo describiremos el procedimiento necesario para la visualización, el registro de nuevas sesiones y la impresión, pues estas difieren con las anteriores. Para listar las sesiones se sigue una secuencia operativa muy similar a la que sigue el listado de tareas. Por su parte, la edición de sesiones supone visualización y luego proceder del mismo modo que para el registro.

En primer lugar, se encuentra el diagrama que representa la visualización de una sesión, figura 4.19. A continuación, en la figura 4.20, se muestra el diagrama que recoge las relaciones del proceso de registro de sesiones. Tras estos, en la figura 4.21, se puede observar el diagrama de secuencia necesario para imprimir sesiones de entrenamiento.

## **Administrador**

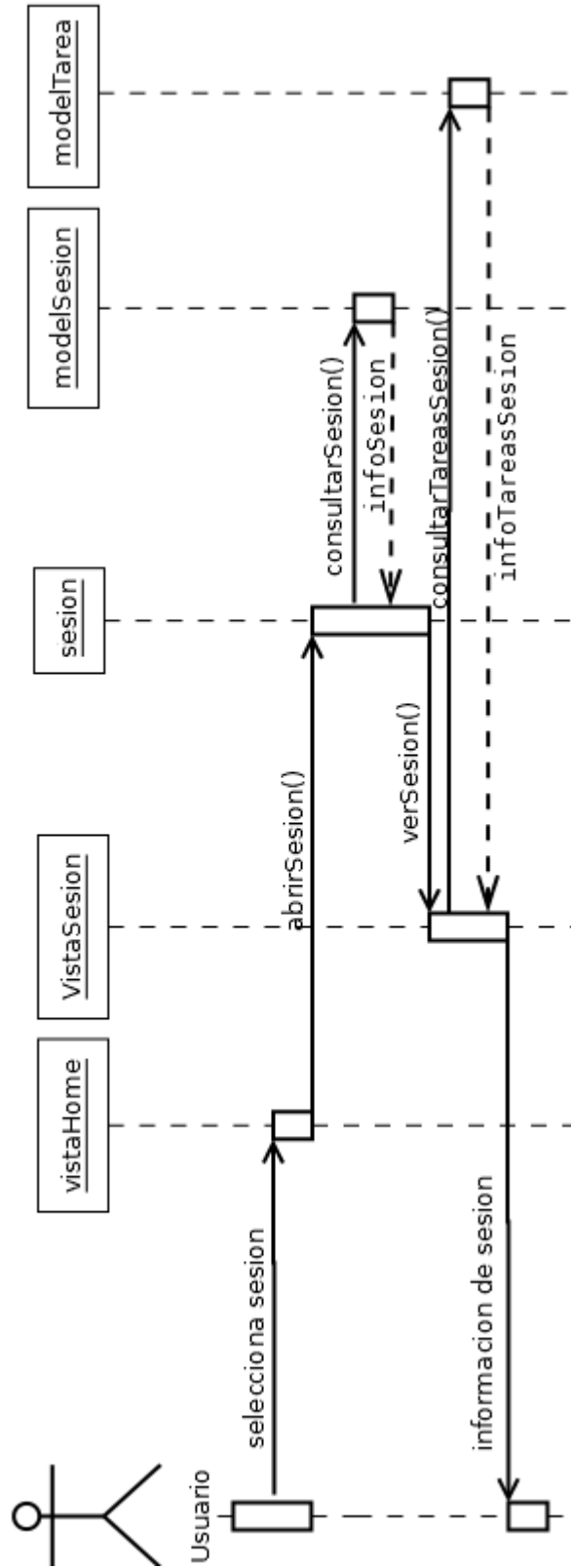
El paquete funcional del administrador permite ver estadísticas y operar con las cuentas de los usuarios. Por tanto, los diagramas de secuencia que se muestran describen las relaciones entre los componentes del sistema para estos procedimientos.

La figura 4.22 muestra la secuencia de procesamiento necesaria para mostrar cualquier estadística del sistema al administrador. Por su parte, en la figura 4.23 se puede observar el modo de procesamiento que requiere modificar las cuentas de usuario, ya sea para bloquearlo, eliminarlo o convertirlo en administrador.

## **Ayuda**

Este paquete ofrece información al usuario para ayudar a la comprensión de la aplicación. Por tanto, el único diagrama de secuencia que habrá que describir corresponde a la visualización de la ayuda, figura 4.24.

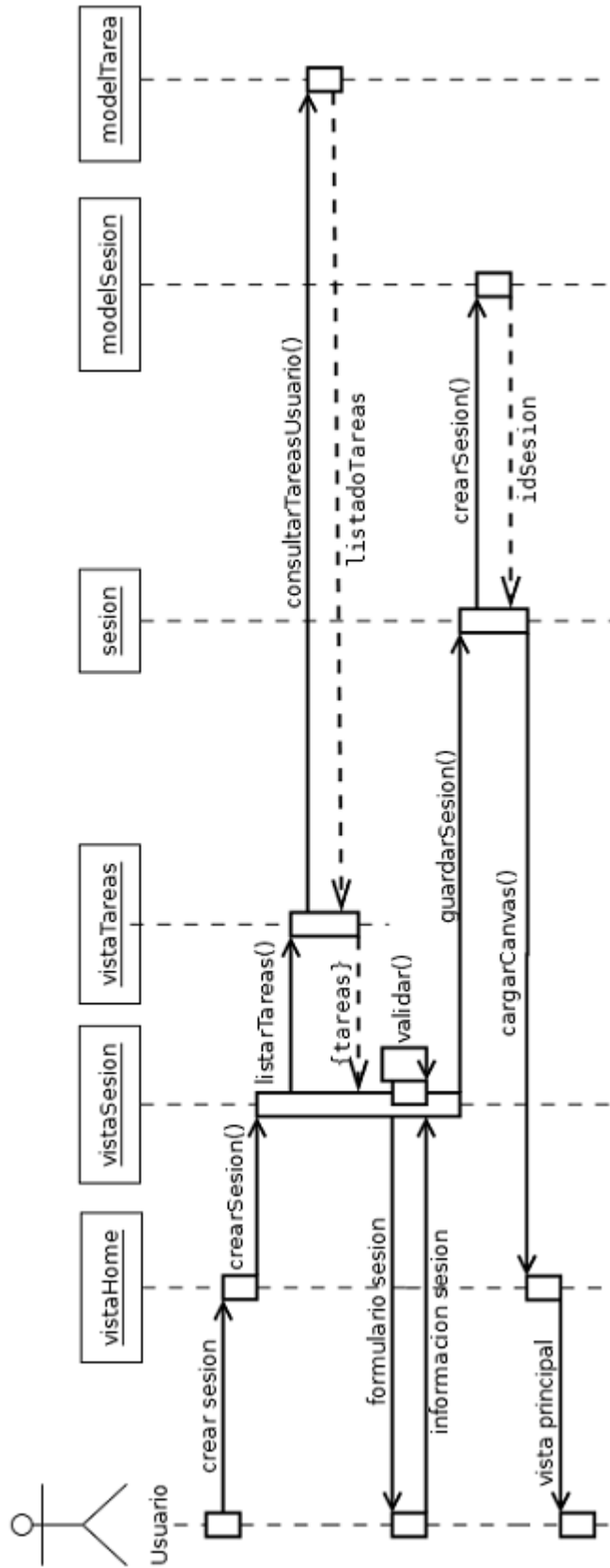
Figura 4.19: Visualizar sesión



*infoTareasSesion* contiene la información del conjunto de tareas incluidos en la sesión de entrenamiento.

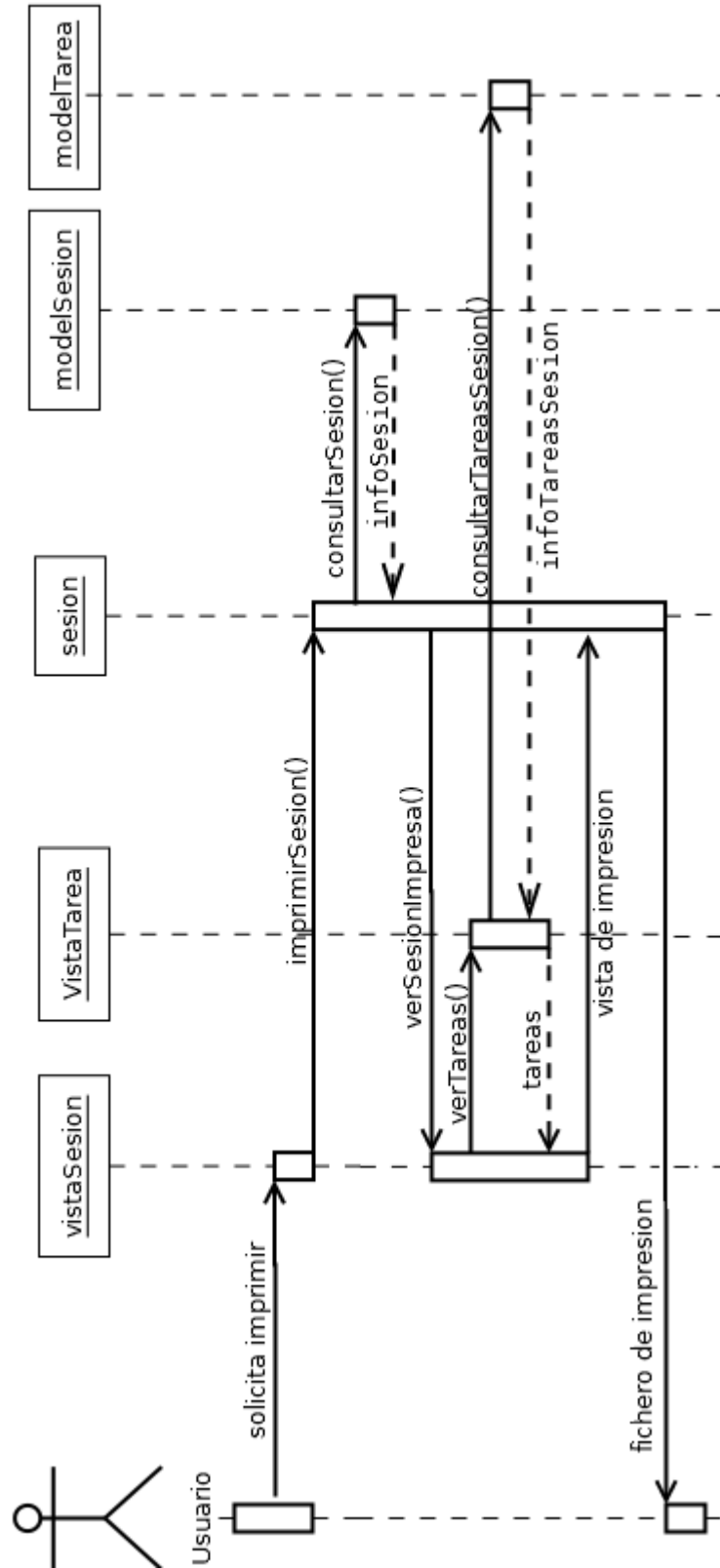


Figura 4.20: Guardar sesión



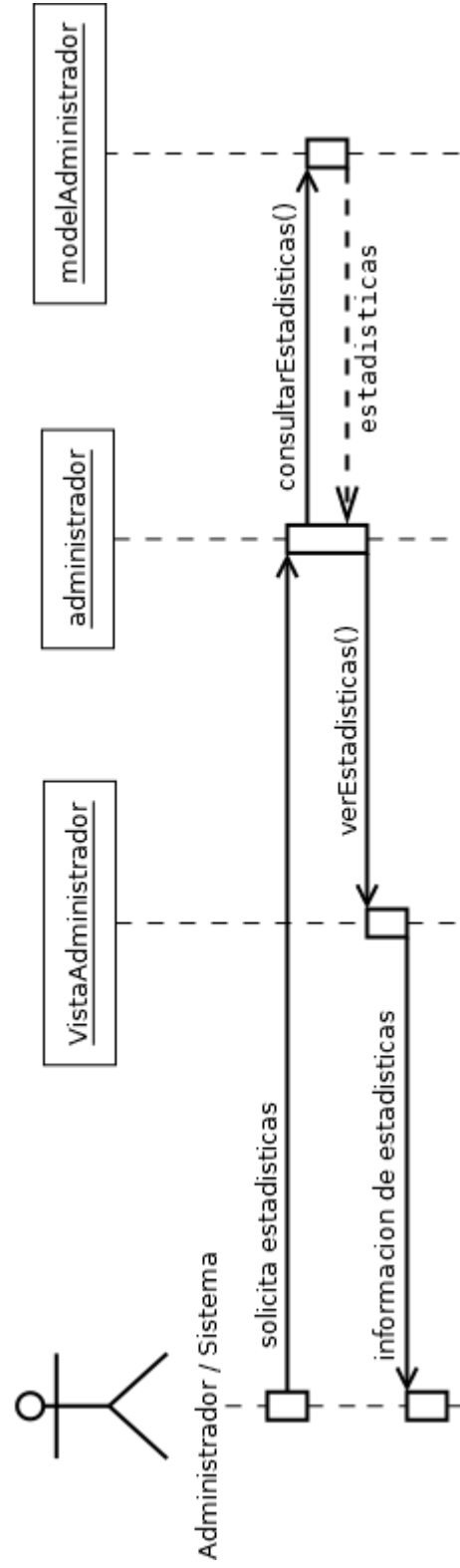
*listadoTareas* contiene la relación de tareas creadas por el usuario para incluir las que desee en la sesión.

Figura 4.21: Imprimir sesión



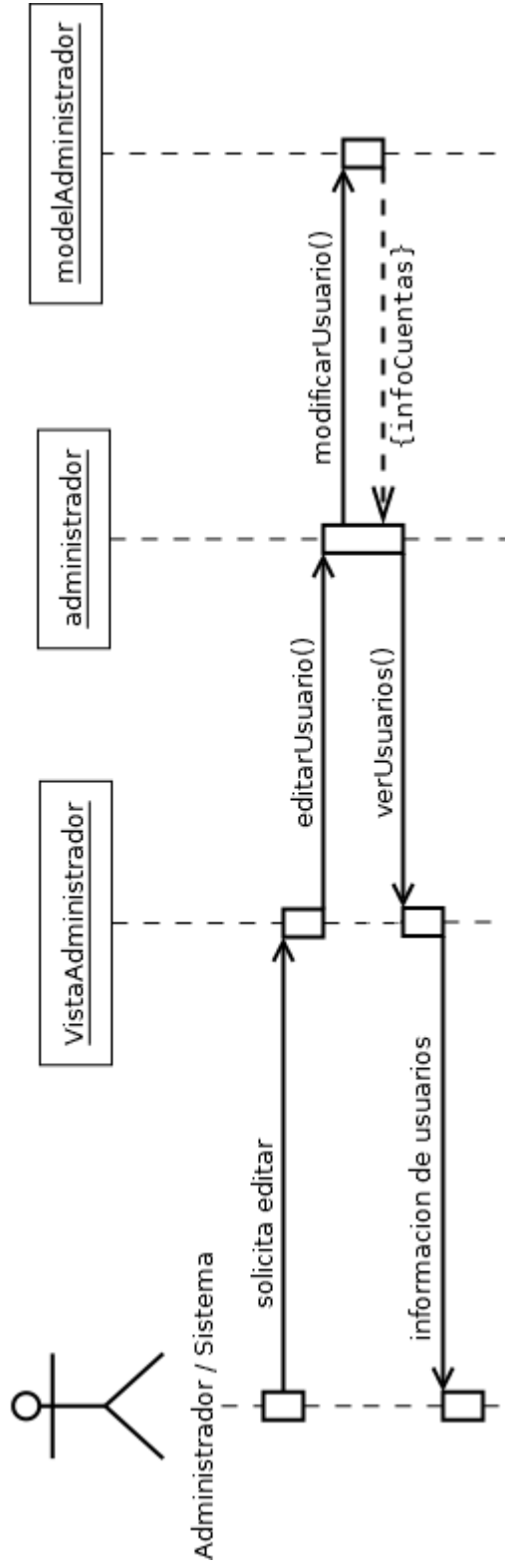
El controlador interactúa directamente con el usuario, ya que debe generar un fichero de impresión.

Figura 4.22: Visualizar estadísticas



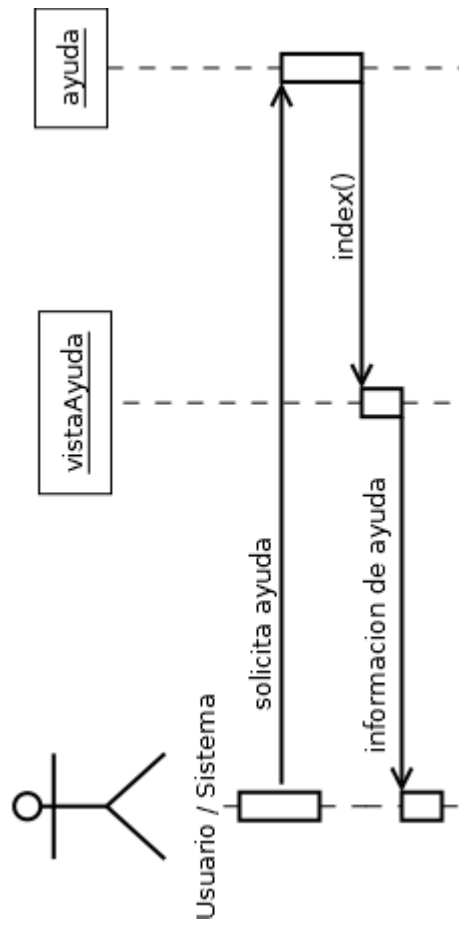
*consultarEstadísticas()* es una función genérica que puede consultar distintos tipos de estadísticas.

Figura 4.23: Modificar cuentas



*editarUsuario* modifica cierta información de la cuenta del usuario, es decir, puede sustituirse por *eliminarUsuario()*, *bloquearUsuario()* o *hacerAdministrador()*.

Figura 4.24: Visualizar ayuda





## Capítulo 5

# Implementación del sistema

### 5.1. Entorno tecnológico

Para escribir los ficheros de código en cualquiera de los lenguajes de programación que se utilicen durante el desarrollo se utilizará *Notepad++*, un editor de texto y código fuente libre que soporta varios lenguajes de programación.

Para el desarrollo de la aplicación utilizaremos *PHP* como lenguaje de programación, en la versión 5.3.1. Se hará uso de *HTML* para el diseño de la interfaz de usuario (vistas) y, concretamente *HTML5*. Por su parte, para la interacción con el usuario se utilizará el lenguaje *JavaScript*.

Además del entorno, vamos a apoyarnos en un framework de *PHP*, *CodeIgniter* en su versión 2.1.3 que es un producto de licencia libre. De este modo el código estará organizado y será más fácil de crear y mantener. *CodeIgniter* se basa en el proceso de desarrollo llamado “Model View Controller” (*MVC* o Modelo Vista Controlador), que es un estándar de programación de aplicaciones, utilizado tanto para hacer sitios web como programas tradicionales. Esta organización concuerda con el modelo de capas de la aplicación propuesto en la subsección 4.1.3 Arquitectura de diseño.

Como framework del lenguaje *JavaScript* se utilizará *Jquery*, un conjunto de librerías gratuitas para cualquier plataforma. De este modo, simplificará la labor de programar los scripts y nos asegurará compatibilidad con los navegadores mas usuales.

Para acceder, mantener y manipular la información del sistema se ha elegido el *SGBD MySQL*. Con el objetivo de hacer mas visual la labor del desarrollador se utilizará la herramienta libre escrita en *PHP*, *phpMyAdmin*. Con la cual se permite manejar la administración de *MySQL* a través de Internet.

Los diagramas *UML* que se han ido utilizando principalmente durante la fase de diseño del sistema, han sido realizados con la herramienta *ArgoUML*. Se trata de una aplicación de diagramado *UML* desarrollada en *Java* y que posee licencia *BSD*. Por su parte, para llevar a cabo el diseño de la interfaz se ha utilizado *Evolus Pencil*, que se trata de software de código abierto que permite realizar wireframes, exportarlos a diferentes formatos de archivo y enlazar las diferentes páginas para simular el flujo de navegación.

Para controlar las versiones hemos elegido *Subversion*, un sistema de control de versiones de software libre bajo licencia de tipo *Apache/BSD*. Se le conoce también como “svn” por ser el nombre de la herramienta utilizada en la línea de comandos. Como interfaz de *Subversion*

hemos instalado *TortoiseSVN*<sup>1</sup>. Se trata del cliente *Subversion*, implementado como una extensión al *shell* de *Windows*, mas popular para este sistema operativo. Por supuesto, es software libre, liberado bajo la licencia *GNU GPL*. Y ademas de esto, un repositorio ofrecido por la herramienta *Redmine*<sup>2</sup>, donde el tutor puede seguir la evolución.

## 5.2. Código fuente

El desarrollo del sistema se ha llevado a cabo utilizando el framework *CodeIgniter* de *PHP*. Por tanto, todo el código fuente se encontrara dentro del sistema de ficheros proporcionado por éste. De tal forma que se describirán con detalle las carpetas, subcarpetas y ficheros para los que se ha escrito código. Por otra parte, aquellas que son propias de *CodeIgniter* únicamente se nombrarán.

```
/CodeIgniter
|-> /application
|   |-> /controllers
|       |-> home.php
|       |-> login.php
|       |-> tarea.php
|       |-> estrategia.php
|       |-> club.php
|       |-> sesion.php
|       |-> administrador.php
|       |-> ayuda.php
```

Cada uno de los ficheros nombrados contiene las funciones necesarias para poner en funcionamiento las vistas relacionadas con el componente correspondiente del sistema y procesar la información relativa a estos.

```
|   |-> /Models
|       |-> model_usuario.php
|       |-> model_tarea.php
|       |-> model_estrategia.php
|       |-> model_club.php
|       |-> model_sesion.php
|       |-> model_administrador.php
|       |-> model_campograma.php
```

Los ficheros del modelo permiten interactuar con la base datos. De modo que, cada uno de los ficheros anteriores contendrá funciones que permiten llevar a cabo las consultas necesarias para cada componente.

---

<sup>1</sup>TortoiseSVN: <https://clinkerpfc.uca.es/svn/entrefutsal>

<sup>2</sup>Redmine: <https://clinkerpfc.uca.es/redmine/projects/entrefutsal>



```

|      |-> /Views
|      |   |-> /Home
|      |   |-> /Perfil
|      |   |-> /Login
|      |   |-> /Tarea
|      |   |-> /Estrategia
|      |   |-> /Club
|      |   |-> /Sesion
|      |   |-> /Administrador
|      |   |   |-> /Estadisticas
|      |   |   |   |-> /Elementos
|      |   |   |   |-> /Generales
|      |   |-> /Ayuda

```

Dentro esta carpeta nos encontramos con una serie de subcarpetas, que contienen los ficheros vista de cada componente del sistema. Estos ficheros contienen la información que interpretan los navegadores, es decir, código *HTML* con *PHP* embebido.

La subcarpeta *Administrador* esta subdividida internamente debido al alto número de ficheros vista de este componente. Esto significa que hemos agrupado los ficheros para conseguir una organización comprensible.

```

|      |   |-> /Cache
|      |   |-> /Config
|      |   |-> /Core
|      |   |-> /Errors
|      |   |-> /Helpers
|      |   |-> /Hooks
|      |   |-> /Language
|      |   |-> /Libraries
|      |   |-> /Logs
|      |   |-> /Third-party
|
|      |-> /css
|      |   |-> /bootstrap
|      |   |   |-> /css
|      |   |   |   |-> bootstrap-flat.css
|      |   |   |-> /js
|      |   |   |   |-> bootstrap.min.js

```

*Bootstrap* es un framework para las hojas de estilo en cascada, *CSS*, que también utiliza *JavaScript* para mejorar la interfaz. Por tanto, */css* contiene el código que ofrece *Bootstrap* con ciertas modificaciones para nuestra aplicación y */js* contiene el código *JavaScript* propio del framework.

```

|      | -> jquery-smoothness.css Este fichero contiene una plantilla que da for-
|      | mato al calendario de sesiones de la aplicación.
|      | -> home.css Contiene clausulas necesarias para adecuar el diseño de la vista
|      | principal del sistema.
|      | -> login.css En este fichero se incluyen clausulas que hacen que las vistas
|      | del componente login sean apropiadas.
|      | -> imprimir.css Formatea la vista de las sesiones que serán proporcionadas
|      | al usuario en forma de ficheros de impresión.
|
| -> /js
|      | -> /canvas
|      | -> configuracion Fichero de propiedades del canvas.
|      | -> main Inicializa los elementos del canvas para dar ofrecer funciona-
|      | lidades y dar el aspecto de pizarra.
|      | -> campograma Contiene la clase campograma, es decir, atributos y
|      | funciones de los pistas de fútbol sala.
|      | -> conector
|      | -> linea
|      | -> movil
|      | -> recta

```

Cada fichero implementa una clase de elementos del campograma.

```
-> /script
-> admin
-> estadisticasAdmin
-> inicioAdmin
-> funciones
-> scriptAdmin
-> scriptCalendario
-> scriptCampograma
-> scriptClub
-> scriptLateral
-> scriptPerfil
-> scriptSesion
-> scriptUsuario
```

Estos ficheros contienen funciones para inicializar las vistas de los componentes a los que corresponden y para complementar la interacción entre estas y el usuario.

```

|
|-> /js-core
|   |-> jquery.js Framework de JavaScript que permite escribir código JavaS-
cript fácilmente.
|   |-> kinetic.min.js Fichero JavaScript de la biblioteca que facilita la explo-
tación del canvas.
|   |-> jquery-ui-datepicker-es.js Plugin de jQuery que permite visualizar
un calendario y seleccionar fechas.
|   |-> /js-color La subcarpeta js-color contiene el fichero JavaScript que per-
mite seleccionar un color para asignárselo a un club y ciertos archivos de imagen para su
interfaz.
|   |-> canvas.js Biblioteca que permite generar gráficas para el administrador.
|
|-> /System
|-> /User-guide

```



## Capítulo 6

# Pruebas del sistema

### 6.1. Pruebas unitarias

Las pruebas unitarias tienen por objetivo localizar errores en cada nuevo artefacto software desarrollado, antes que se produzca la integración con el resto de artefactos del sistema.

Por tanto, durante la implementación del código fuente se han ido realizando pruebas unitarias de forma manual a los ficheros de cada componente. A continuación se describe como se han realizado las pruebas en función del tipo de fichero:

**Scripts** Basándonos en *firebug* de *Mozilla Firefox* se han observado las variables de cada script para detectar posibles errores o incongruencias.

**Modelo** Se han desarrollado funciones en el controlador para ejecutar cada una de las funciones de los ficheros modelo. De forma que se pudiera comprobar a través de *phpMyAdmin* que los trasposos de información son fiables.

**Vista** Se han cargado cada una de las vistas en el navegador y si ha sido necesario se ha modificado tanto los ficheros vista como los ficheros *css* del sistema.

### 6.2. Pruebas de integración

Este tipo de pruebas tienen por objetivo localizar errores en módulos o subsistemas completos, analizando la interacción entre varios artefactos software. De modo que, las pruebas de integración de este sistema han sido realizadas tras la implementación de cada nuevo artefacto y una vez probado individualmente éste mismo. El modo de ejecución de estas pruebas ha sido la ejecución del sistema, probando cada funcionalidad del nuevo componente, basándose en las funciones del controlador.

**Controlador** Las funciones de los controladores requieren ejecutar los modelos, vistas y scripts del nuevo componente y de los anteriores. Por tanto, probar un nuevo controlador supone realizar pruebas incrementales de integración del sistema.

## 6.3. Pruebas del sistema

En esta actividad se realizan las pruebas de sistema de modo que se asegure que el sistema cumple con todos los requisitos establecidos entre el cliente y el desarrollador. Este proceso se puede dividir en dos partes bien diferenciadas, donde se verifique por un lado la funcionalidad del sistema y por otro los requisitos complementarios, también conocidos como requisitos no funcionales.

### 6.3.1. Pruebas funcionales

Con estas pruebas se analiza el buen funcionamiento de la implementación de los flujos normales y alternativos de los distintos casos de uso del sistema. Dicha verificación se ha basado en los requisitos descritos en la sección 3.2 Requisitos funcionales.

A modo de descripción del trabajo realizado con las listas de comprobación se describirán algunos casos de prueba relevantes de la aplicación.

#### Identificar usuario

A continuación se describe el caso de prueba para el flujo normal de la identificación de un usuario y el flujo alternativo respectivamente.

Tabla 6.1: Caso de prueba: Identificación de usuario correcta

CP-001	Identificación de usuario correcta
<b>Descripción</b>	Un usuario registrado introduce información correcta para identificarse.
<b>Precondiciones</b>	El usuario está registrado e introduce la información adecuadamente.
<b>Resultado esperado</b>	El usuario obtiene acceso a la aplicación.
<b>Evaluación</b>	El sistema actúa adecuadamente.

Tabla 6.2: Caso de prueba: Identificación de usuario errónea

CP-002	Identificación de usuario errónea
<b>Descripción</b>	Un usuario introduce información inadecuada para identificarse.
<b>Precondiciones</b>	El usuario no está registrado o introduce información errónea.
<b>Resultado esperado</b>	El usuario permanece en la pantalla de identificación y se notifica el error.
<b>Evaluación</b>	El sistema actúa adecuadamente.

Análogamente a estos casos de prueba, se ha verificado la funcionalidad del registro de nuevos usuarios o la recuperación de cuentas. Considerándose innecesarios de describir por su similaridad.

#### Registrar tarea

El registro de tarea supone la verificación mas compleja del paquete funcional de tareas, ya que pueden darse mas flujos alternativos que para el resto de los casos de uso del componente.

Tabla 6.3: Caso de prueba: Registro de tarea

<b>CP-003</b>	<b>Registro de tarea</b>
<b>Descripción</b>	Un usuario solicita almacenar una tarea e introduce información necesaria.
<b>Precondiciones</b>	El usuario está identificado e introduce la información adecuadamente.
<b>Resultado esperado</b>	La tarea queda almacenada y aparece en el listado de tareas.
<b>Evaluación</b>	El sistema actúa adecuadamente.

Tabla 6.4: Caso de prueba: Registro de tarea sin objetivos

<b>CP-004</b>	<b>Registro de tarea para usuario sin objetivos</b>
<b>Descripción</b>	Un usuario solicita almacenar una tarea sin haber registrado objetivos.
<b>Precondiciones</b>	El usuario está identificado y no existen objetivos.
<b>Resultado esperado</b>	El usuario puede crear el objetivo mientras rellena la información de la tarea.
<b>Evaluación</b>	El sistema actúa adecuadamente.

Tabla 6.5: Caso de prueba: Registro de tarea incorrecto

<b>CP-005</b>	<b>Registro de tarea con información errónea</b>
<b>Descripción</b>	El usuario almacena una tarea pero no introduce información adecuada.
<b>Precondiciones</b>	El usuario está identificado.
<b>Resultado esperado</b>	El sistema notifica el error al usuario y permanece en la misma página.
<b>Evaluación</b>	El sistema actúa adecuadamente.

Las estrategias se han probado del mismo modo que las tareas. De forma que no será necesario detallar estos casos de prueba, pues se consideran similares a los del paquete funcional descrito anteriormente.

Por otra parte, la gestión de clubes y de planificación proponen casos de uso similares a los anteriores pero aun mas sencillos. Por tanto, aunque se ha verificado su correcta funcionalidad no incluiremos los casos de prueba de estos paquetes funcionales.

## Visualizar estadísticas

El administrador tiene acceso a varias estadísticas del uso de la aplicación, ya sea para ver los usuarios registrados, clubes, elementos del campograma, etc... El siguiente caso de prueba es genérico para cualquiera que sea la información solicitada por el administrador.

Tabla 6.6: Caso de prueba: Visualizar estadísticas

<b>CP-006</b>	<b>Visualizar estadísticas</b>
<b>Descripción</b>	El administrador solicita visualizar información estadística.
<b>Precondiciones</b>	El administrador está identificado.
<b>Resultado esperado</b>	El sistema muestra información tabulada y gráfica para su fácil comprensión.
<b>Evaluación</b>	El sistema actúa adecuadamente.

### 6.3.2. Pruebas no funcionales

Estas pruebas pretenden comprobar el funcionamiento del sistema, con respecto a los requisitos no funcionales definidos en la etapa de análisis.

#### Portabilidad

La verificación de la portabilidad del sistema se ha llevado a cabo de dos maneras. Por un lado, se ha probado el sistema en varios navegadores a través de la herramienta *Browserstack* nombrada en la sección del capítulo anterior. Además de esto, en la medida de lo posible, se ha probado en varios navegadores a través de distintos dispositivos físicos. Es decir, se ha comprobado que la aplicación permite ser utilizada bien desde PCs con diferentes sistemas operativos, o bien en dispositivos táctiles también bajo distintos sistemas operativos.

**Browserstack** Para ver la compatibilidad del software con los distintos navegadores se ha utilizado *Browserstack*<sup>1</sup>. Se trata de una interfaz web comercial que permite ejecutar cualquier aplicación web en distintos navegadores y versiones. Además de esto, es posible testear aplicaciones que no se encuentren alojadas en ningún servidor de Internet, indicándole la carpeta local que debe ejecutar.

Al tratarse de una web comercial no hemos podido utilizar por completo su funcionalidad, ya que ofrece un plan gratuito que permite usar la interfaz durante una hora. No obstante, ha sido suficiente para testear los navegadores mas comunes en sus versiones modernas y comprobar que el sitio se ejecuta con normalidad.

**modern.IE** *Internet Explorer*, el navegador de *Microsoft*, presenta múltiples problemas de compatibilidad con los estándares propuestos por el *W3C*<sup>2</sup>. Esta es una organización independiente y neutral que desarrolla estándares relacionados con la Web. Por esta razón, *Microsoft* ha lanzado un servicio para testear gratuitamente sitios web, *modern.IE*<sup>3</sup>

Para nuestra aplicación hemos generado un informe que se puede observar en el Anexo I. Tras la lectura del reporte se han llevado a cabo las acciones necesarias para mejorar las sugerencias que se han considerado apropiadas. Cabe destacar que el contenido del reporte no muestra importantes problemas de compatibilidad con el navegador.

#### Seguridad

Las pruebas realizadas al software han verificado que el sistema cumple los requisitos de seguridad establecidos en la etapa de análisis.

**Login de usuario** Todo usuario que pretenda acceder al sistema debe identificarse con su email y una contraseña que el mismo ha establecido. Esta contraseña ha sido encriptada con uno de los métodos ofrecidos por el framework *CodeIgniter*.

---

<sup>1</sup>Browserstack: <http://www.browserstack.com>

<sup>2</sup>W3C: <http://www.w3.org>

<sup>3</sup>modern.IE: <http://loc.modern.ie/es-es>



**Acceso restringido** Un usuario solo puede tener acceso a la información que él mismo haya registrado. Por ello el sistema utiliza relaciones entre las diferentes tablas de la base de datos que le permita conocer el propietario de cada registro.

## Fiabilidad

El sistema ofrece ventanas de confirmación para las funciones que puedan suponer pérdidas de información. De esta forma, se requiere confirmación del usuario para los procedimientos que se han considerado de riesgo.

## Extensibilidad

El diseño arquitectónico del sistema se ha basado en el patrón Modelo-Vista-Controlador y el desarrollo que se ha llevado a cabo ha sido un proceso incremental. Por estas dos razones, podemos asegurar que cumple con los requisitos establecidos de extensibilidad.

## Escalabilidad

Para asegurar la escalabilidad, se ha implementado la base de datos en un SGBD fiable, MySQL. De este modo, si el volumen de usuarios crece en gran medida, no supondrá un problema a la base de datos.

Por otro lado, a pesar de que el plan de hosting contratado es gratuito, éste no supone grandes limitaciones en cuanto al espacio de disco ni a la transferencia de datos. No obstante, si fuese necesario escalar el sistema, no significaría un proceso demasiado laborioso debido a la facilidad para contratar un plan de hosting más competitivo.

## Rendimiento

Para verificar el rendimiento del sistema se han utilizado bibliotecas que ofrecieran funcionalidad eficiente y fiable. No obstante, con el objetivo de verificar externamente la aplicación se ha utilizado la herramienta *GTmetrix*<sup>4</sup>, cuyo objetivo es medir la velocidad de sitios web.

El resultado del análisis de rendimiento de nuestra aplicación puede observarse en el Anexo II. Cabe destacar que el contenido del reporte califica nuestra aplicación satisfactoriamente. No obstante, algunos aspectos no son bien valorados y, por ello se describen a continuación:

- **Inline small CSS:** Se sugiere incluir código *css* en línea. Bajo nuestra responsabilidad, esta no se considera una buena práctica.
- **Enable Keep Alive:** Debido al plan contratado para el servidor este aspecto es innegociable.
- **Use a Content Delivery Network (CDN):** Los recursos actuales del sistema no nos permite mejorar en este sentido.
- **Configure entity tags (ETags):** Estas etiquetas son fácilmente configurables en Apache para páginas estáticas.

---

<sup>4</sup>GTmetrix: <http://gtmetrix.com/>

- **Use cookie-free domains:** Ofrecer esta funcionalidad mejoraría el rendimiento de los recursos estáticos como imágenes pero requiere un subdominio donde alojar estos recursos.

## 6.4. Pruebas de aceptación

Las pruebas de aceptación la han realizados usuarios potenciales del sistema. Con la intención de conocer la opinión que de verdad interesa, la de las personas que realmente lo utilizarán. Estas opiniones se han recogido como informes que se incluyen el Anexo III. Los usuarios seleccionados para realizar estas pruebas compiten en distintas modalidades y categorías a nivel regional y nacional.

En conclusión a la lectura de los informes del anexo, se puede afirmar que existe un buen grado de aceptación por parte de los futuros usuarios del sistema.

Parte III

Epílogo



## Capítulo 7

# Manual de usuario

### 7.1. Introducción

El producto software desarrollado es una aplicación web para entrenadores de fútbol sala que tiene como principal objetivo la planificación de sesiones de entrenamiento. Estas sesiones poseen información acerca de ciertos conceptos como la fecha, la hora y el lugar o el club y el responsable de dirigirlas y, por otra parte, se componen de una serie de tareas que han de ser realizadas. Por tanto, antes de planificar una sesión, el usuario deberá registrar las tareas que desea incluir y el club que llevará a cabo la sesión.

Como se ha mencionado anteriormente, se trata de una aplicación web. Por tanto, es posible acceder a ella con gran facilidad. Cualquier usuario que pretenda acceder al sistema únicamente deberá introducir la *URL* en su navegador:

<http://trainingfs.x10.mx>

### 7.2. Características

A continuación se nombran las principales funcionalidades del sistema:

#### **Edición de campograma**

El sistema proporciona una campograma que representa una pista de fútbol sala y una serie de elementos que podrán ser introducidos en este como jugadores, porterías o líneas entre otros. Esta pista puede ser visualizada en dos formatos, media pista o pista completa. La edición del campograma supone representar una situación específica en la pista que simule una tarea o una estrategia.

#### **Gestión de tareas y estrategias**

Las tareas representan los ejercicios que se realizan durante las sesiones de entrenamiento. Por su parte, las estrategias simulan situaciones de juego que se pretenden poner en práctica durante los partidos. Tanto las tareas como las estrategias están compuesta de

una descripción textual y un campograma. La funcionalidad que el usuario puede realizar con ambas son almacenarlas, visualizarlas, editarlas, eliminarlas y enumerarlas en un listado.

A diferencia de las tareas, las estrategias no pueden ser incluidas en las sesiones.

## **Gestión de clubes**

El sistema permite gestionar clubes por los usuarios para personalizar sus sesiones. El usuario que registra un club ha de introducir como mínimo su nombre pero puede contener información adicional como un escudo, una foto de plantilla o un color representativo. La gestión de clubes incluye crear clubes, editarlos, eliminarlos, visualizarlos y enumerarlos.

## **Gestión de planificación**

La planificación se refiere a la gestión de sesiones de entrenamiento. Una sesión está compuesta de información descriptiva de la sesión y una relación de tareas a realizar. Entre las funcionalidades de la gestión de planificación se encuentra crearlas, editarlas, eliminarlas, visualizarlas y mostrarlas en el calendario. Además de estas, existe una funcionalidad que se considera principal en el sistema, imprimir la sesión, ya que produce un documento que el usuario portará al entrenamiento para guiarse durante la sesión.

## **7.3. Requisitos previos**

Un navegador actual es el único requisito del sistema, ya que el sistema emplea ciertas funcionalidades de *HTML*, el lenguaje de los navegadores en su versión mas moderna, HTML5.

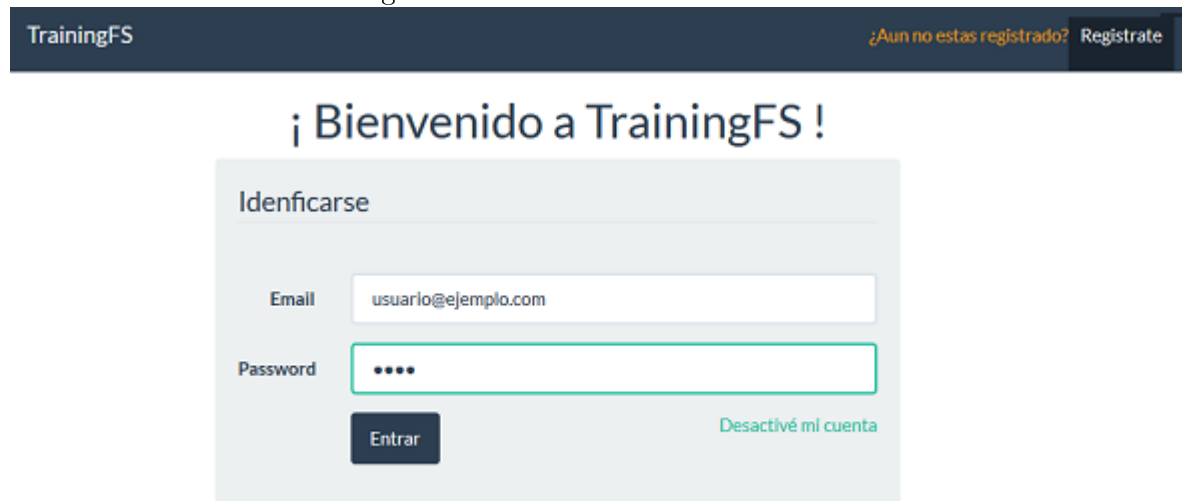
## **7.4. Utilización**

En esta sección se detalla la secuencia de pasos que el usuario debe seguir para realizar las funcionalidades principales.

### **Acceso al sistema**

1. Abra el navegador e introduzca la *URL* del sistema: <http://trainingfs.x10.mx>
2. La secuencia depende si el usuario está registrado en el sistema o pretende acceder por primera vez.
  - a) Los usuarios registrados deben introducir el email y la contraseña que lo identifican en el formulario que se muestra en la figura 7.1.

Figura 7.1: Identificación de usuario



TrainingFS [¿Aun no estas registrado?](#) [Registrate](#)

## ¡ Bienvenido a TrainingFS !

### Identificarse

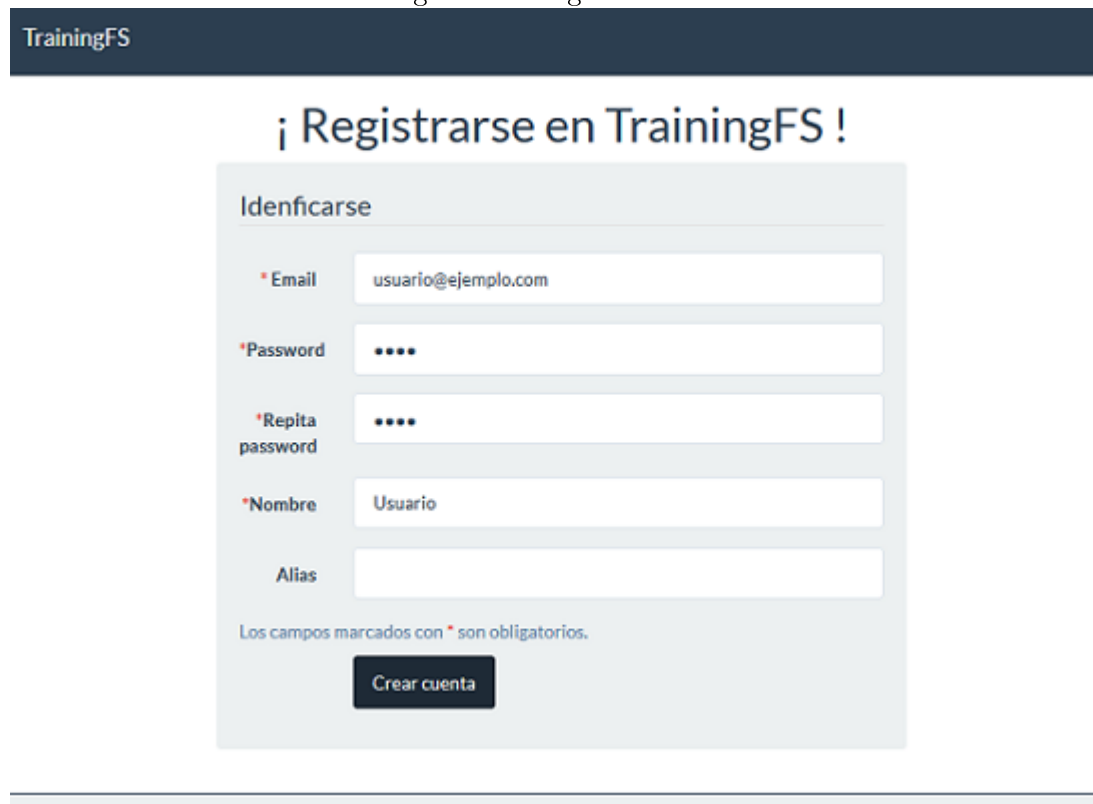
Email

Password

[Desactivé mi cuenta](#)

- b) Los nuevos usuarios deberán seleccionar “Registrate” para ser redirigidos y rellenar el formulario de registro que se observa en la figura 7.2.

Figura 7.2: Registro de usuario



TrainingFS

## ¡ Registrarse en TrainingFS !

### Identificarse

\* Email

\* Password

\* Repita password

\* Nombre

Alias

Los campos marcados con \* son obligatorios.

3. Si rellenan los campos solicitados correctamente, se obtiene acceso al sistema.

Figura 7.3: Acceso al sistema



## Registrar objetivos

Los objetivos sirven para clasificar las tareas según la utilidad para las que estas se desarrollen. Es decir, un objetivo llamado "Calentamiento" contendrá las distintos ejercicios que el entrenador utilice para calentar.

1. Introducir el nombre del objetivo en el campo que se encuentra bajo el listado y seleccionar "Crear" como se muestra en la figura 7.4.



Figura 7.4: Registrar objetivo

Tareas Estrategias Clubs

No existen objetivos definidos por el usuario.

Las tareas son clasificadas por objetivos como "Finalización", "Velocidad", "Presión", etc...

Para crear objetivos solamente introduzca el nombre en el campo que aparece abajo.

Calentamiento Crear

Observe que el objetivo creado aparece en el listado de tareas.

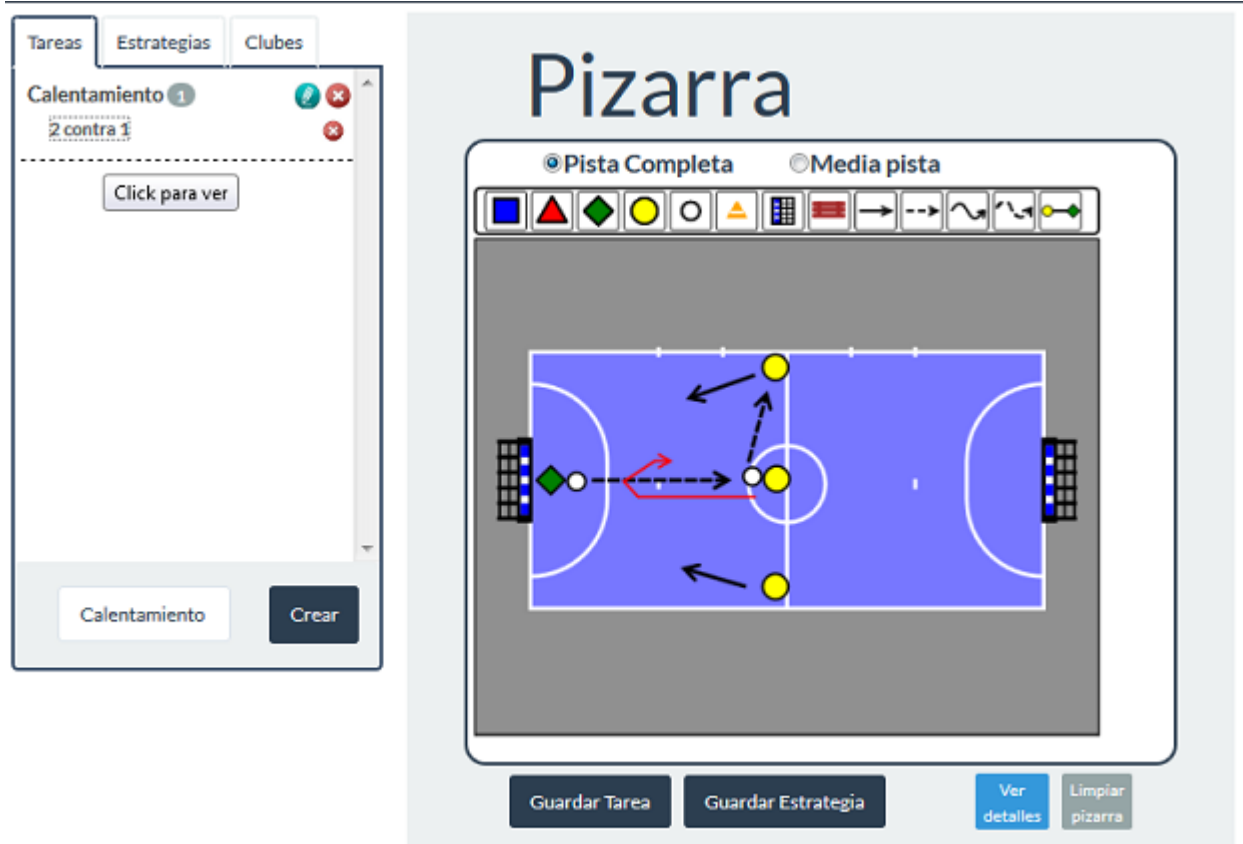
Tanto el registro de acciones como de clubes se realiza del mismo modo que el descrito para los objetivos. La única apreciación que se debe tener en cuenta radica en que ha de seleccionarse la pestaña correspondiente entre las situadas en la parte superior del listado.

### Edición de tareas

Vamos a detallar la edición de tareas, pues así se describirán dos funcionalidades, visualización en primer lugar y, posteriormente se operará del mismo modo que para el registro.

1. Para visualizar una tarea en el campograma, únicamente es necesario seleccionarla en el listado de tareas que aparece a la izquierda del campograma, figura 7.5.

Figura 7.5: Visualizar tarea



Por otra parte, es posible observar también la descripción textual de la tarea haciendo clic en el botón “Ver detalles” situado bajo el campograma.

2. A partir de aquí, es un proceso similar al registro. Primeramente se editará el campograma según sea necesario y, luego, se hará clic en el botón “Guardar Tarea” colocado bajo el campograma.
3. Introducir la información textual de la tarea en el formulario que aparecerá en el lugar del campograma y que se muestra en la figura 7.6. Una vez introducida la información se requiere hacer clic en “Crear nueva” o “Sobrescribir” según convenga.

Figura 7.6: Guardar tarea

El formulario 'Guardar tarea' está estructurado de la siguiente manera:

- Nombre:** Campo de texto con el valor '2 contra 1'.
- Objetivo:** Selector de lista desplegable con el valor 'Calentamiento'.
- Organización:** Campo de texto con el valor 'Los tres jugadores de campo se colocan en la línea divisoria de medio campo.'
- Desarrollo:** Campo de texto con el valor 'El portero saca a cualquiera de los tres jugadores. El que recibe el balón pasa a un lado y se coloca para defender.'
- Reglas:** Campo de texto con el valor 'Los atacantes deben dar al menos un pase al compañero antes de finalizar la jugada.'
- Periodización:** Campo de texto con el valor 'En cuanto termine la acción el portero sacará a otros tres jugadores que estarán colocados de la misma manera que los anteriores.'

En la parte inferior del formulario hay tres botones: 'Crear nueva', 'Sobreescribir' y 'Cancelar'.

En el listado de tareas podrá observarse que la tarea ha sido guardada.

Las estrategias disponen de la misma operabilidad que las tareas. Los usuarios encontrarán diferencias únicamente en el formulario de la descripción textual.

## Registro de sesión

1. Para crear una nueva sesión de entrenamiento el usuario debe pinchar el botón “Planificar sesión” con lo que aparecerá un formulario en el lugar donde se encontraba el campograma, figura 7.7.

Figura 7.7: Registro sesión

# Planificar sesión

## Información de la sesión

\* Fecha

2014-03-11

\* Hora

19:15

Fin

21:00

Entrenador

Sergio Barroso

\* Club

Cádiz CF - Virgili F

Semana

25

Sesión

Microciclo

Intenso

Cádiz CF - Virgili FS

Caden

Observaciones

Pabellon Ciudad de Cadiz.

## Tareas de la sesión

Tareas

Calentamiento

2 contra 1

Presión

Finalización

Sesión

2 contra 1

Salida de presión 4 vs 3

Presión - control - disparo

Click para incluir en la sesión

Guardar sesión

Cancelar

Marzo 2014

Lu	Ma	Mi	Ju	Vi	Sa	Do
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Planificar sesión

- El usuario debe introducir la información de la sesión y seleccionar las tareas que desea realizar durante la sesión. Para ello tendrá que buscarla en el listado izquierdo, pinchar sobre ella y aparecerá en el listado de la sesión. Para este ejemplo se han incluido tres tareas en la sesión como se puede apreciar en la figura 7.7. Si la información introducida por el usuario es correcta, el sistema mostrará la sesión y marcará el día en el calendario.

## Imprimir sesión

La impresión de una sesión supone generar un documento *pdf* que contenga la información de la sesión. Esta funcionalidad exige que se visualice la sesión y, posteriormente se solicite la impresión.

1. Para visualizar una sesión tan solo es necesario hacer clic en el día del calendario para el que se ha planificado la sesión. En la figura 7.8 se muestra el calendario con el día 11 de Marzo marcado, haciendo clic sobre éste se abre la visualización de la sesión en el lugar del campograma como también se puede observar en la figura.

Figura 7.8: Visualizar sesión

**Sesión 11/03/2014**

**Información**

Hora inicio: 19:15      Hora fin: 21:00      Semana: 25  
Club: Cadiz CF - Virgili FS      Entrenador: Sergio Barroso  
Microciclo: Intenso  
Lugar: Pabellon Ciudad de Cadiz.

**Tareas**

2 contra 1  
Organización: Los tres jugadores de campo se colocan en la línea divisoria de medio campo.

Imprimir

Editar    Eliminar    Cerrar

**Marzo 2014**

Lu	Ma	Mi	Ju	Vi	Sa	Do
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

Planificar sesión

En la figura anterior puede observarse la visualización de la sesión, tanto su información como sus tareas.

2. Los botones situados bajo la visualización de la sesión permiten operar con ella. El botón “Imprimir” genera automáticamente un documento *pdf*.

## Capítulo 8

# Manual de instalación

### 8.1. Introducción

Se ha desarrollado una aplicación web para entrenadores de fútbol sala que pretende ayudar a todo tipo de técnicos a diseñar sus entrenamientos de manera fácil e intuitiva. Por ello, consideramos importante el hecho de no requerir instalación en el lado del cliente. Es decir, el usuario no necesita incluir en su ordenador personal nada que no tenga cualquiera. Los únicos requisitos son un navegador y acceder a la *URL* del sistema:

<http://trainingfs.x10.mx>

De esta forma, el peso de la instalación ha recaído en el lado del servidor.

### 8.2. Requisitos previos

Un usuario tan solo necesita un navegador para poder acceder al sistema, siendo éste el único requisito para el uso de la aplicación. No obstante, el campograma del sistema está implementado con una etiqueta de *HTML5*, “canvas” que tan solo es compatible con navegadores actuales. Por tanto, la compatibilidad con esta etiqueta representará la compatibilidad con el sistema. A continuación se enumeran la primeras versiones de los navegadores más comunes para la que se obtuvo una versión totalmente compatible:

- *Internet Explorer 9.0* y su versión móvil *IE Mobile 10.0*
- *Firefox 26.0*
- *Chrome 31.0*
- *Safari 7.0* y su versión móvil, *iOS Safari 3.2*
- *Opera 19.0*, su versión móvil *Opera Mini* no dispone de una versión totalmente compatible
- *Android Browser 3.0*
- *Blackberry Browser 7.0*

### 8.3. Procedimientos de instalación

Anteriormente hemos comentado que por parte del usuario no se requiere ningún tipo de instalación. Para ello se ha conseguido que al sistema se acceda a través de su *URL*. Por tanto, a continuación se describe el proceso de instalación en el lado del servidor. Cabe mencionar que *x10hosting* ha sido el servicio seleccionado para alojar nuestro sistema. Por otro lado, debemos mencionar que contamos con todos los ficheros de código fuente y un fichero *sql* con la estructura de la base de datos.

1. Registrarse como usuario en *x10hosting* contratando el plan gratuito, el cual contiene ciertas restricciones que no limitan nuestro sistema. Para ello se requiere introducir un correo electrónico válido y una contraseña para acceder al gestor que proporciona.
2. Crear el sitio web con el instalador de software que ofrecen. Esta opción facilita en gran medida el proceso de instalación, ya que tan solo hemos tenido que seleccionar *CodeIgniter* entre los frameworks que proponen y el servidor nos ha proporcionado la estructura de carpetas y ficheros que componen el framework.
3. Crear la base de datos con el instalador de MySQL que el sistema ofrece. En nuestro caso, se ha exportado la base de datos creada para el servidor local utilizado durante el desarrollo. Debemos destacar que el sistema ofrece *phpMyAdmin* para hacer este trabajo mas llevadero.
  - a) Consultar los detalles de instalación para obtener el nombre de la base de datos que el instalador ha creado.
  - b) Seleccionar la pestaña “Databases” del menú superior y, luego abrir la herramienta *phpMyAdmin*.
  - c) Seleccionamos la base de datos que ha creado el instalador y pinchar el botón “Import” que está situado en el menú superior.
  - d) Seleccionamos el archivo *trainingfs.sql* situado en “/CodeIgniter/trainingfs.sql” y pinchamos en “Go”. El resto de opciones las mantenemos con el valor por defecto.
  - e) Una vez ejecutada con éxito la operación, tendremos la base de datos con la estructura definida en la figura 4.8.
4. Modificar los archivos de configuración para personalizar el sistema. Es decir, hemos definido nuestra base de datos, seleccionado las librerías que se cargan por defecto, etc... A partir de aquí, debemos acceder al gestor de archivos de *x10hosting*. Para ello seleccionaremos “My Websites” en el menú superior y abriremos la herramienta *File Manager*.
  - a) Ajustar la *URL* base de la aplicación:
    - 1) Abrir el fichero ubicado en “/public\_html/application/config/config.php”.
    - 2) Editar el valor de la variable `$config['base_url']` y guardar el fichero.

En nuestro caso, el servidor local contiene:

```
$config['base_url'] = "http://localhost/TrainingFS/CodeIgniter/";
```

Para el servidor de *x10hosting* hemos definido:

```
$config['base_url'] = "http://trainingfs.x10.mx/";
```

b) Asegurar que la base de datos está definida correctamente:

- 1) Abrir el fichero ubicado en `"/public_html/application/config/database.php"`.
- 2) Comprobar el valor de las variables que aparecen a partir de la línea 50, especialmente `$db['default']['database']`. Esta variable debe contener entre comillas el nombre de la base de datos creada anteriormente. Estos valores deben ser proporcionados por el sistema. En nuestro caso, debemos cotejar con los siguientes valores:

```
$db['default']['hostname'] = 'localhost';  
$db['default']['username'] = 'web';  
$db['default']['password'] = 'usuarioweb';  
$db['default']['database'] = 'trainingfs';  
$db['default']['dbdriver'] = 'mysql';
```

c) Definir las herramientas del framework que se cargarán automáticamente:

- 1) Abrir el fichero que está en `"/public_html/application/config/autoload.php"`.
- 2) Buscar la variable `$autoload['libraries']` e indicar qué librerías serán cargadas por defecto. Es decir, para nuestro sistema la definiríamos así:

```
$autoload['libraries'] = array('database', 'session', 'dompdf_lib');
```

5. Subir los ficheros de código que componen el sistema. Esta labor ha sido la más tediosa, ya que se han de subir manualmente tanto los ficheros del modelo, controlador y la vista como los scripts, hojas de estilo y bibliotecas externas.

a) Subir los archivos PHP de la aplicación, `"/public_html/application"`, para el modelo, la vista y el controlador.

- 1) Abrir la carpeta `"/public_html/application/models"`, pinchar en el menú superior "Upload" y seleccionar para subir cada uno de los ficheros alojados en `"/CodeIgniter/application/models"`.
- 2) Abrir la carpeta `"/public_html/application/views"` y crear subcarpetas seleccionando "New folder" en el menú superior con los mismos nombres que encontramos en `"/CodeIgniter/application/views"`. Tras esto, subir todos los ficheros que contiene cada subcarpeta pinchando en "Upload" y seleccionando cada fichero de la subcarpeta.
- 3) Abrir la carpeta `"/public_html/application/controllers"`, pinchar en el menú superior "Upload" y seleccionar para subir cada uno de los ficheros alojados en `"/CodeIgniter/application/controllers"`.



- b) Subir los archivos *CSS* del sistema.
  - 1) Crear en `"/public_html"` una carpeta llamada `"/css"` pulsando "New folder" en el menú superior. Del mismo modo, dentro de ella crearemos otras dos carpetas `"/images/"` y `"/bootstrap/"`. En la segunda de estas subcarpetas, la que contendrá el framework *Bootstrap*, crearemos otras tres carpetas llamadas `"/css"`, `"/fonts"` y `"/js"`. Con esto conseguiremos que la estructura de la carpeta `"/public_html/css"` sea similar a `"/CodeIgniter/css"`.
  - 2) Subir seleccionado "Upload" cada fichero situado en `"/CodeIgniter/css"` y en las subcarpetas de esta según correspondan.
- c) Subir las bibliotecas de *JavaScript* necesarias para el sistema.
  - 1) Crear en `"/public_html"` una carpeta llamada `"/js-core"` pulsando sobre "New folder" en el menú superior. Del mismo modo, dentro de ella crearemos una subcarpeta llamada `"/jscolor"` para que la estructura sea idéntica a `"/CodeIgniter/js-core"`.
  - 2) Subir cada fichero que se encuentra en `"/CodeIgniter/js-core"` seleccionado "Upload" y los que se ubican en `"/CodeIgniter/js-core/jscolor"`, de esta forma conseguiremos que `"/public_html/js-core"` disponga del mismo contenido.
- d) Subir los ficheros de *JavaScript* desarrollados para la aplicación.
  - 1) Crear en `"/public_html"` una carpeta llamada `"/js"` pulsando "New folder" en el menú superior. Del mismo modo, dentro de ella crearemos dos subcarpeta llamadas `"/script"` y `"/canvas"` para que la estructura sea idéntica a `"/CodeIgniter/js"`.
  - 2) Subir seleccionado "Upload" cada fichero que se encuentra en cada subcarpeta de `"/CodeIgniter/js"` para que `"/public_html/js"` disponga del mismo contenido que su homólogo.
- e) Subir la librería *DomPDF* de *PHP*.
  - 1) Crear con "New folder" toda la estructura de carpetas y subcarpetas en `"/public_html/application/libraries"`, del mismo modo que se puede observar en `"/CodeIgniter/application/libraries"`.
  - 2) Subir con "Upload" cada fichero ubicado en `"/CodeIgniter/application/libraries"` y en las subcarpetas para que `"/public_html/aplication/libraries"` contenga los mismos archivos.

## 8.4. Pruebas de implantación

Este conjunto de pruebas engloba un proceso incremental paralelo al desarrollo que se ha realizado antes de entregar cada versión a los usuarios. Para cada incremento del sistema se ha actualizado la aplicación en el servidor en busca de errores de cualquier tipo. Los problemas detectados han sido solucionados asignándoles máxima prioridad, de forma que se pudiera ofrecer una versión consistente del software.

Para ello, durante las primeras versiones el desarrollador ha sido el encargado de probar cada funcionalidad del sistema y cotejar la información que éstas producían. Así, para cada iteración se repetía el proceso desde distintas cuentas de usuario y dispositivos, con

diferentes conexiones a Internet y variando la franja horaria con el objetivo de verificar la fiabilidad del software y la comunicación con la base de datos.

Una vez desarrollada la primera versión final del producto, se solicitó a varios usuarios que se registraran en el sistema, accedieran y probaran las funcionalidades a su antojo. Este proceso fue supervisado por el desarrollador, que mantuvo contacto con ellos mientras hacían uso de la aplicación para cotejar la información que introducían, con los registros de la base de datos. Para este chequeo de información se usaron dos vías, *phpMyAdmin* y el panel de administrador que incluye el sistema.

Los principales aspectos en los que este proceso ha sido determinante están relacionados con la compatibilidad, ya que para beneficio del sistema y del usuario, se disponía de distintos dispositivos y navegadores. Además, cabe destacar que en escasas ocasiones ha sido problemático el plan contratado para alojar el sistema. Por este motivo, se considera lo suficiente fiable para mantener actualmente la aplicación bajo estos servicios.

## Capítulo 9

# Conclusiones

### 9.1. Objetivos

El sistema desarrollado permite al usuario planificar sus sesiones de entrenamiento e imprimir un documento que le sirva de guía para la realización de esta. Este documento contiene la descripción gráfica y textual del conjunto de tareas que se pretenden llevar a cabo e información de la sesión del tipo fecha, hora, lugar, etc...

Como se ha mencionado, las sesiones están compuestas de información propia de ellas y una serie de tareas que previamente han sido registradas por el usuario. Por tanto, podemos afirmar que el sistema dispone de un gestor de tareas que permite almacenarlas incluyéndoles información textual y gráfica. Estas descripciones se complementan para proporcionar una buena comprensión de las tareas. De forma que, la información textual está dividida en secciones tales como el nombre, el objetivo para el que se realiza y secciones del modo de ejecución. Por su parte, la descripción gráfica se lleva a cabo con un campograma que trata de emular a las pizarras propias de los entrenadores, conteniendo gran variedad de funcionalidades.

Paralelamente a las tareas y de modo similar se permite al usuario diseñar estrategias y almacenarlas en el sistema. Igualmente son descritas tanto gráfica como textualmente.

Al margen de estas, pueden numerarse funcionalidades de uso relacionadas con el perfil del usuario y la gestión de clubes. La primera de estas, permite al usuario editar su información con el objetivo de personalizar su interfaz. Por otra parte, la gestión de clubes tiene como objetivo ofrecer al usuario la posibilidad de crear los clubes para los que trabaja y, así personalizar sus sesiones de entrenamiento.

### 9.2. Lecciones aprendidas

A nivel personal, la lección más valiosa de este proyecto radica en la confianza obtenida para realizar un proyecto que satisfaga necesidades reales de los usuarios. Hasta la fecha no había realizado ningún proyecto más allá de los propuestos en las asignaturas de la titulación, para los que no me había encontrado más dificultades que la propia programación. Al afrontar un proyecto real surgen problemas de distintos orígenes, ya sea por el desconocimiento de las tecnologías o de los lenguajes. Es por esto, que ahora considero las metodologías de ingeniería del software imprescindibles para estructurar la realización de

proyectos.

En cuanto a la formación continua puedo afirmar que las comunidades de Internet son una herramienta muy útil a la hora de encontrar soluciones a problemas que surgen durante los proyectos. El hecho de mantener contacto con personas involucradas en el mismo tipo de tareas que uno mismo, permite extraer conocimientos de las experiencias comunes. Por tanto, en el futuro seguiré formando parte de estas comunidades, buscando respuestas y colaborando con ellas.

Aunque el desarrollo de este proyecto no se ha realizado propiamente en equipo, he descubierto un conjunto de herramientas que hace la colaboración mucho mas llevadera. Me estoy refiriendo a los gestores de tareas y repositorios, utilidades que no conocía hasta el momento de comenzar este proyecto. Debido al fuerte hincapié para que trabajase con ellas que ha hecho el coordinador del proyecto, Iván Ruiz Rube, he podido valorar la utilidad que suponen. Pues, mantener el estado de las tareas asignadas a cada miembro y actualizar el trabajo realizado permite a cualquier equipo de proyecto conocer el estado del mismo con gran facilidad.

Obviamente, un aspecto formativo muy importante ha sido el conocimiento de varios lenguajes de programación y el Sistema de Gestión de Base de Datos, *MySQL*. Entre los lenguajes cabe destacar *PHP*, *HTML*, *JavaScript* y *CSS*. Para cada uno de ellos se ha requerido un aprendizaje y en la medida de lo posible se han utilizado frameworks que han permitido que la curva de aprendizaje fuese creciente.

No son de menos importancia las herramientas utilizadas durante todas las etapas del proyecto. Las distintas fases de la ingeniería del software han requerido el aprendizaje de herramientas de diagramado *UML* y de un editor de textos de Latex, *LyX*, para la documentación de esta memoria. He de decir que estas herramientas también han supuesto un proceso de toma de contacto para poder explotar adecuadamente sus funcionalidades.

### 9.3. Trabajo futuro

Este proyecto se ha basado en los requisitos de un usuario concreto, Sergio Barroso. Por tanto, aunque se espera buena aceptación entre los entrenadores de fútbol sala, es posible que en futuras versiones se consideren nuevos requisitos o se editen algunos ya propuestos. Para ellos se valorará el uso de los usuarios del sistema y las sugerencias que estos propongan. A continuación se describen funcionalidades que pueden ser incluidas en próximas versiones del proyecto:

**Adecuación del sistema para otros deportes** Con el objetivo de ampliar el número de usuarios se baraja la opción de modificar el campograma para que entrenadores de otros deportes hagan uso de nuestro sistema. No obstante, esta ampliación puede requerir la modificación de otras funcionalidades. Para ello, se tratará de concretar reuniones con entrenadores de diferentes deportes como fútbol, baloncesto o balonmano que nos permita reescribir los requisitos funcionales.

**Fondo de campograma personalizado** Este adecuación permitiría al usuario subir una imagen mas acorde a sus necesidades. De este modo, un usuario que disponga de un archivo que considere apropiado para sí mismo podrá utilizarlo como fondo de pista.

**Buscador de clubes** Este requisito permitiría al sistema no duplicar información de los clubes. Es decir, dos entrenadores que hayan entrenado el mismo club no supondría crear dos clubes. Cabe mencionar que tanto el desarrollador del proyecto como el coordinador consideran esta una funcionalidad apropiada para la primera versión. No obstante, el usuario que ha ejercido como cliente insiste en que los distintos entrenadores no deben compartir información bajo ningún concepto.

**Tareas animadas** Aunque esta funcionalidad nos alejaría del objetivo principal que es generar ficheros de impresión, posiblemente facilitaría la comprensión de estas. Para valorar si es preferible este formato valoraremos la opinión de los usuarios durante la explotación del sistema.

# Bibliografía

- [1] Booch, G.: *El lenguaje unificado de modelado*. Pearson Educación (2006)
- [2] Robert, C.: *Código limpio: Manual de estilo para el desarrollo ágil de software*. Anaya Multimedia (2012)
- [3] Ullman, L.: *PHP paso a paso*. Anaya Multimedia (2009)
- [4] Dubois, P.: *La biblia de MySQL*. Pearson Educación (2009)
- [5] Gauchat, J.: *El gran libro de HTML5, CSS3 y JavaScript*. Marcombo (2012).
- [6] Manual oficial de CodeIgniter (Enero 2014),  
<http://ellislab.com/codeigniter/user-guide>
- [7] Manual oficial de ArgoUML (Febrero 2014),  
<http://argouml-stats.tigris.org/documentation/manual-0.34>
- [8] Manual oficial de MySQL (Diciembre 2013),  
<http://dev.mysql.com/doc/refman/5.0/es/>
- [9] Manual oficial de phpMyAdmin (Diciembre 2013),  
<https://phpmyadmin-spanish.readthedocs.org/en/latest/user.html>
- [10] Sitio web oficial del consorcio W3C (Febrero 2014), <http://www.w3c.org>
- [11] Sitio web oficial de PHP (Febrero 2014), <http://php.net/docs.php>
- [12] Sitio web oficial de jQuery (Febrero 2014), <http://jquery.com>
- [13] Sitio web oficial de Bootstrap (Febrero 2014), <http://getbootstrap.com>
- [14] GNU General Public License (Marzo 2014),  
<http://www.gnu.org/copyleft/gpl.html>
- [15] Sitio web oficial del Software Lyx (Marzo 2014), <http://www.lyx.org>
- [16] Sitio web oficial de XAMPP (Marzo 2014), <http://www.apachefriends.org>

## Anexo I - Informe de *modern.IE*

*modern.IE* es una herramienta de análisis para sitios web bajo el punto de vista de la compatibilidad con el navegador de Microsoft, Internet Explorer. Desde su sitio web hemos podido obtener un reporte de la calidad del código fuente, el cual es adjuntado a continuación:



## Resultados de la página para: <http://trainingfs.x10.mx/>

modern.IE site scan for  
common coding issues



Get screenshots of your site  
across browsers and devices



Scan for code no longer  
supported in modern IE



**¿En qué consiste esta herramienta?** En la actualidad, hacer que tus experiencias web sean geniales (o que, simplemente, funcionen) en diversos navegadores y dispositivos puede llevar mucho tiempo. Esta herramienta detecta los procedimientos de codificación comunes que pueden conllevar problemas de compatibilidad o impedir que los usuarios disfruten de la mejor experiencia posible con la página web. Siempre que resulte posible, recomendamos aplicar correcciones o mejoras de acuerdo con estándares web como HTML5 y CSS3 (o con procedimientos alternativos adecuados). No se trata de una lista de comprobación completa para la codificación de Web actual, sino de un punto de partida que te permite emplear menos tiempo en la realización de pruebas para IE y más tiempo en la creación de lo que de verdad importa en Web.

Este escáner se encuentra en versión (compilación 30 de sept de 2013). [Envíanos tus comentarios](#) para ayudarnos a mejorar modern.IE.

### Resultados del análisis



#### Corrije problemas comunes que surgen de la compatibilidad con versiones anteriores de IE



##### Problemas de compatibilidad conocidos

No hemos detectado problemas de compatibilidad conocidos relativos a la representación de esta página web en Internet Explorer. Puede deberse a que no realizamos tests de todas las páginas que componen un sitio web. Te sugerimos que utilices [Compat Inspector](#) para detectar automáticamente problemas de compatibilidad comunes cuando efectúes tests para Internet Explorer tanto si detectamos problemas como si no.

No se sugiere ninguna acción adicional.



##### Modo de compatibilidad

No hemos detectado que esta página web se represente en modo de compatibilidad. Esto significa que se representa en [modo estándar](#), lo que puede mejorar el rendimiento de la página web en Internet Explorer 9 y 10.

No se sugiere ninguna acción adicional.



##### Marcos y bibliotecas

Hemos detectado que [los marcos y las bibliotecas](#) de esta página web son compatibles con las últimas versiones de Internet Explorer o que los marcos y las bibliotecas no se encuentran en esta página web.

No se sugiere ninguna acción adicional.





Hemos detectado que esta página web contiene una declaración [que reconoce estándares web actuales como HTML5 y CSS3](#). Esto significa que esta [página web se representa en modo estándar](#) en Internet Explorer 9 y 10. Al representarse en este modo, la página web puede alcanzar un rendimiento superior en Internet Explorer.

No se sugiere ninguna acción adicional.



## Contribuye a que esta página web funcione correctamente en diversos navegadores y dispositivos



### Prefijos de CSS

#### Mejora sugerida

#### ¿Qué hemos detectado?

Hemos detectado que esta página web puede no contener [prefijos específicos de proveedores](#) o puede haber implementado prefijos específicos de proveedores cuando no son precisos en las propiedades de CSS comunes. Esto puede provocar problemas de compatibilidad relativos a la representación de esta página web en Internet Explorer o en otros exploradores actuales.

Este informe incluye propiedades de CSS comunes que provocan problemas de compatibilidad y que, en la actualidad, son transiciones, animaciones y transformaciones.

Cuando resulta posible, este informe también contiene los prefijos de proveedores que representan numerosos navegadores, entre los que se incluyen Chrome (-webkit), Firefox (-moz), Internet Explorer (-ms), Opera (-o) y Safari (-webkit).

Ayúdanos a mejorar modern.IE: [envíanos tus comentarios](#).

#### ¿Por qué es importante?

Puede que sus usuarios no disfruten de la mejor experiencia que ofrece esta página web en navegadores actuales como Internet Explorer 9 o 10.

Queremos ayudarte a llegar al mayor número de usuarios posible con la mejor experiencia de navegación. El 21,3 % de los usuarios realiza visitas con Internet Explorer 9 y 10 en equipos de escritorio según [NetApps](#) (diciembre de 2012).

Las prácticas recomendadas en materia de prefijos específicos de proveedores también pueden ayudar a sus usuarios a disfrutar de la mejor experiencia en dispositivos como el iPad de Apple, Microsoft Surface y el Kindle de Amazon, así como con otros navegadores actuales en pantallas más reducidas. Estas experiencias con dispositivos móviles representan alrededor del [12 % del tráfico global de Internet](#), cifra que se espera que aumente rápidamente en los próximos 5 años (Business Insider, noviembre de 2012).

#### Implementación:

[Conocer las prácticas recomendadas relativas a los prefijos de proveedores](#)

[Ver la compatibilidad con CSS en IE](#)

#### Test it across browsers:



mediante BrowserStack en un par de clics



mediante la descarga de una máquina virtual

A continuación, se presenta una lista de selectores y de sus propiedades que pueden causar problemas de compatibilidad.

#### Ubicación:

<http://trainingfs.x10.mx/css/bootstrap/c...>

Selector	Propiedad	Número de la línea de origen
.progress-striped	-webkit-linear-gradient	5776
.progress-bar		
.progress-striped	-webkit-linear-gradient	5791
.progress-bar-success		
.progress-		

Selector	Propiedad	Número de la línea de origen
striped .progress-bar-info	-webkit-linear-gradient	5800
.progress-striped .progress-bar-warning	-webkit-linear-gradient	5809
.progress-striped .progress-bar-danger	-webkit-linear-gradient	5818
.carousel-control.left	-webkit-linear-gradient	6809
.carousel-control.right	-webkit-linear-gradient	6816



### Complementos de navegadores

Hemos detectado que esta página web [no contiene complementos](#). Esto contribuye a que sus usuarios disfruten de una mejor experiencia con el sitio en dispositivos como el iPad de Apple, Microsoft Surface y el Kindle de Amazon, así como con otros navegadores actuales sin complementos en pantallas más reducidas.

No se sugiere ninguna acción adicional.



### Diseño web adaptativo

Hemos comprobado que esta página web utiliza [un diseño web adaptativo](#) para ser compatible con varios dispositivos. Esto permite que los usuarios obtengan una mejor experiencia en dispositivos como iPad de Apple, Microsoft Surface, Kindle de Amazon, XBOX 360 y otros exploradores actuales en pantallas más grandes o más pequeñas.

Los puntos de interrupción deben definirse de acuerdo con el contenido de su sitio. Le recomendamos los siguientes puntos de interrupción comunes como punto de partida para escritorios, tabletas y plataformas móviles de destino:

The graph below shows the window widths for which we detected media query support (in green). We recommend testing your site at the non-highlighted window widths to confirm it works well.

NOTE: This graph uses common breakpoints which may help you target specific form-factors.



No se recomienda ninguna otra acción.



### Detección de navegadores

Hemos detectado que esta página web no emplea la detección de navegadores para brindar compatibilidad con los navegadores actuales. Si no ha implementado la detección de características, te sugerimos que lo haga, puesto que puede reducir el número de tests adicionales que se precisan cuando se publican nuevas versiones de los navegadores.

No se sugiere ninguna acción adicional.

## ✓ Optimize the images on your page

Congratulations! Your images cannot be optimized any further or any savings made are insignificant. You are good to go!

No further action is suggested.

## ! HTML5 inputs

Mejora sugerida

¿Qué hemos detectado?

Your page is not using any of the modern HTML5 input types. If you have a form, double check if you can use any of those to improve your users' experience.

Help us improve modern.IE - [Submit feedback](#).

¿Por qué es importante?

Browsers that support the new HTML5 input types can improve the users' experience by providing a keyboard layout better fit to that field (an email for example) or showing a dedicated control (date). This helps your users to avoid mistakes while filling it and to speed up the process.

Corrección:

HTML5 input types in IE11

Test it across browsers:



mediante BrowserStack en un par de clics



mediante la descarga de una máquina virtual

## ! Prerender + prefetch

Mejora sugerida

¿Qué hemos detectado?

You are not using any technique to load content in advance while your users are reading your content and improving your website's performance. Check if using some of these techniques improves performance.

Help us improve modern.IE - [Submit feedback](#).

¿Por qué es importante?

By using these techniques you are providing your users a faster experience while browsing your content and keep them more engaged.

Corrección:

Prerender and prefetch support

Test it across browsers:



mediante BrowserStack en un par de clics



mediante la descarga de una máquina virtual

## ✓ Compressed content

Your server is sending compressed content and your users enjoy a fast browsing experience.

No further action is suggested.

## ⬇ Considera la creación con algunas características nuevas de Windows 8

### ! Valor predeterminado de navegación táctil

Obtener información

¿Qué hemos detectado?

Hemos detectado que esta página web no tiene configurado el comportamiento predeterminado para la navegación táctil en Internet Explorer 10 con dispositivos que ejecutan Windows 8.

Puede configurarlo añadiendo una línea de código, es decir, la propiedad de CSS "-ms-touch-action", como sigue:

```
canvas {  
  
    -ms-touch-action: double-tap  
    -zoom;  
  
}
```

Cada vez es mayor el número de usuarios que navegan en Web mediante dispositivos táctiles. Te sugerimos que configure el comportamiento táctil como predeterminado para la interacción de los usuarios con la página web de manera táctil.

Además, puedes optimizar esta página web para facilitar la navegación táctil y hacer que sea natural. Te sugerimos el empleo de la [API de MSPointers](#) (eventos de puntero) en la página web para brindar compatibilidad con la navegación táctil, además de la navegación con el mouse y el teclado para ofrecer una interacción más avanzada. Hace poco, Microsoft envió la API de MSPointers a W3C con el fin de contribuir a que la Web se encamine hacia la interacción táctil.

Ayúdanos a mejorar modern.IE: [envíanos tus comentarios](#).

¿Por qué es importante?

Puede que los usuarios no disfruten de la mejor experiencia que ofrece esta página web en dispositivos optimizados para la interacción táctil como el iPad de Apple, Microsoft Surface o los equipos con Windows 8. Asimismo, [se espera un aumento rápido](#) en el número de dispositivos móviles con funcionalidades táctiles durante los próximos 5 años (Business Insider, noviembre de 2012).

Implementación:

Configurar el comportamiento táctil predeterminado

Probar la API de MSPointers para la entrada avanzada

Test it across browsers:



mediante BrowserStack en un par de clics



mediante la descarga de una máquina virtual



## Flip Ahead Browsing

Hemos detectado algo

¿Qué hemos detectado?

Your page is not using the "prev" and "next" link relations. These attributes help search engines and browsers better understand the hierarchy of your website content and enable new features to improve the browser experience of your visitors like [Flip ahead browsing](#)

Help us improve modern.IE - [Submit feedback](#).

¿Por qué es importante?

These attributes will not only enable the flip ahead functionality in modern IE (making page navigation a lot easier) but will also help search engines to better understand how your content is organized for SEO purposes.

Corrección:

To enable pagination and Flip Ahead browsing, add the following to your page's <head> element

```
<link rel="next" href="/next  
"/>  
<link rel="prev" href="/prev  
"/>
```

Flip ahead browsing

Test it across browsers:



mediante BrowserStack en un par de clics

## ! IE11 tiles + Notifications

[Obtener información](#)

¿Qué hemos detectado?

We've found that this webpage hasn't built a Windows 8.1 Start Screen tile. A Start Screen tile puts your site logo closer to your users when they can "pin" it and launch this webpage right next to Windows Store apps.

This can be completed with two lines of code and a PNG image like this:

```
<meta name="msapplication-TileColor"
content="#123456" />
<meta name="msapplication-square150x150logo" content="square.png"
"/>
```

If you want to provide more image options for the different tile sizes available in Windows 8.1 go to [www.buildmypinnedsite.com](http://www.buildmypinnedsite.com) to know more.

Also notifications can help you engage more with your users and get more frequent visits.

Help us improve modern.IE - [Submit feedback](#).

¿Por qué es importante?

Your users may not be getting this webpage's best experience on devices like Microsoft Surface or Windows 8.1 PCs. This feature helps them personalize their Windows 8 Start Screen with their favorite site like yours.


Corrección:

[Create a Windows 8.1 tile for your site](#)

[Connect your site to your app](#)

Test it across browsers:

 mediante BrowserStack en un par de clics

 mediante la descarga de una máquina virtual

### Centers

[Dev Center Home](#)

[Windows Store apps](#)

[Internet Explorer](#)

[Desktop](#)

[Hardware](#)

### Related web developer sites

[SkyDrive](#), [Hotmail](#), and [Skype](#)  
[Windows Azure](#)  
[Windows Phone](#)  
[Xbox](#)

### Other Internet Explorer sites

[Enterprise](#)  
[Home users](#)

### Downloads

[Windows 8.1 Preview with IE11](#)  
[IE11 Preview for Windows 7](#)  
[Internet Explorer 10 for Windows 7](#)  
[Internet Explorer 9](#)  
[Internet Explorer samples](#)  
[More downloads](#)

### Extras

[modern.IE](#)  
[Compat Inspector](#)  
[F12 Developer tools](#)

### Stay connected

[IEBlog](#)  
[Exploring IE Blog](#)

### Support

[Forums](#)  
[Top solutions](#)  
[Microsoft community](#)  
[Windows 8 support](#)



## Anexo II - Reporte de *GTmetrix*

GTmetrix es un servicio web que sirve de gran ayuda para detectar los aspectos que hacen más lento cualquier sitio web. Tan sólo requiere introducir la dirección de la web que se pretende analizar y en cuestión de segundos se obtiene un completo informe en el que se analizan muchos factores que afectan a la velocidad de carga. Cada uno de estos aspectos indican varios valores para mejorar la comprensión del reporte.



## Performance Report for: <http://trainingfs.x10.mx/>

Report generated: Monday, March 24, 2014, 5:05 AM -0700

Test Server Region: Vancouver, Canada

Using: Firefox (Desktop) 25.0.1, Page Speed 1.12.16, YSlow 3.1.7

Page Speed Grade:

(86%) ↑ Avg: 79%

B

YSlow Grade:

(94%) ↑ Avg: 79%

A

Page load time: 0.48s | Total page size: 159KB | Total number of requests: 9

### Priority Issues (Top 5)

Inline small CSS	D (67)	↓ Avg Score: 93%	CSS	High
Enable Keep-Alive	C (77)	↓ Avg Score: 96%	Server	High
Specify a cache validator	B (83)	↓ Avg Score: 94%	Server	High
Avoid a character set in the meta tag	B (85)	↓ Avg Score: 95%	Content	High
Avoid CSS @import	B (85)	↓ Avg Score: 96%	CSS	Medium

### How does this affect me?

Studies show that users leave a site if it hasn't loaded in 4 seconds; keep your users happy and engaged by providing a fast performing website.

As if you didn't need more incentive, **Google has announced that they are using page speed in their ranking algorithm.**

### About GTmetrix

We can help you develop a faster, more efficient, and all-around improved website experience for your users. We use Google Page Speed and Yahoo! YSlow to grade your site's performance and provide actionable recommendations to fix these issues.

### About the Developer



GTmetrix is developed by the good folks at **Gossamer Threads**, a Vancouver-based company with over 16 years experience in web technology.  
[www.gossamer-threads.com](http://www.gossamer-threads.com)

### What do these grades mean?

This report is an analysis of your site with Google and Yahoo!'s metrics for how to best develop a site for optimized speed. The **grades you see represent** how well the scanned URL adheres to those rules.

Lower grades (C or lower) mean that the page can stand to be faster using better practices and optimizing your settings.

### What's in this report?

This report covers basic to technical analyses on your page. It is categorized under many headings:

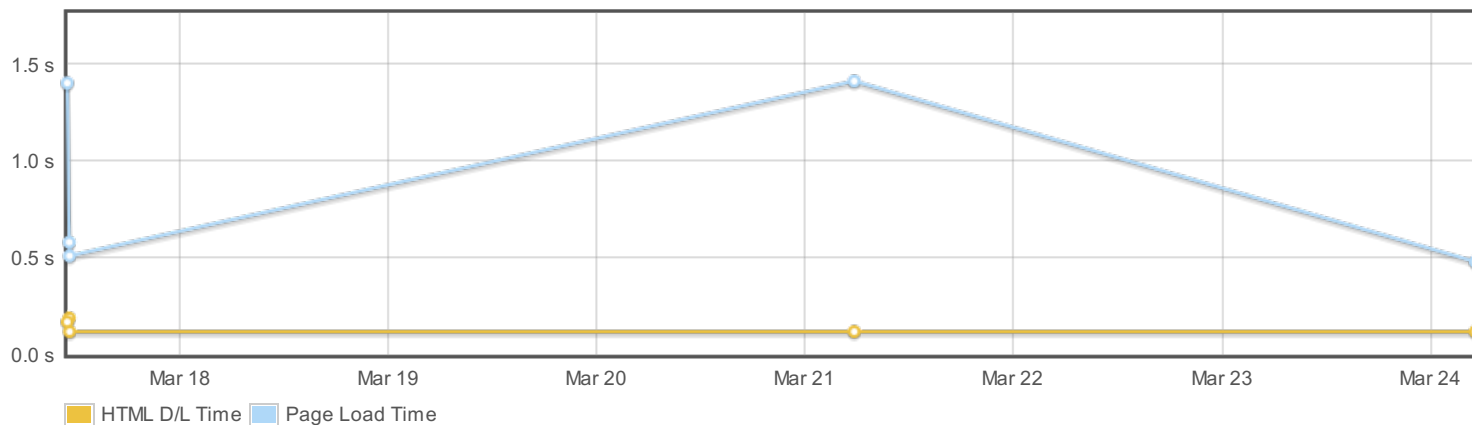
- **Executive:** Overall score information and Priority Issues
- **History:** Graphed history of past performance
- **Waterfall:** Graph of your site's loading timeline
- **Technical:** In-depth Page Speed & YSlow information

These will provide you with a snapshot of your performance.

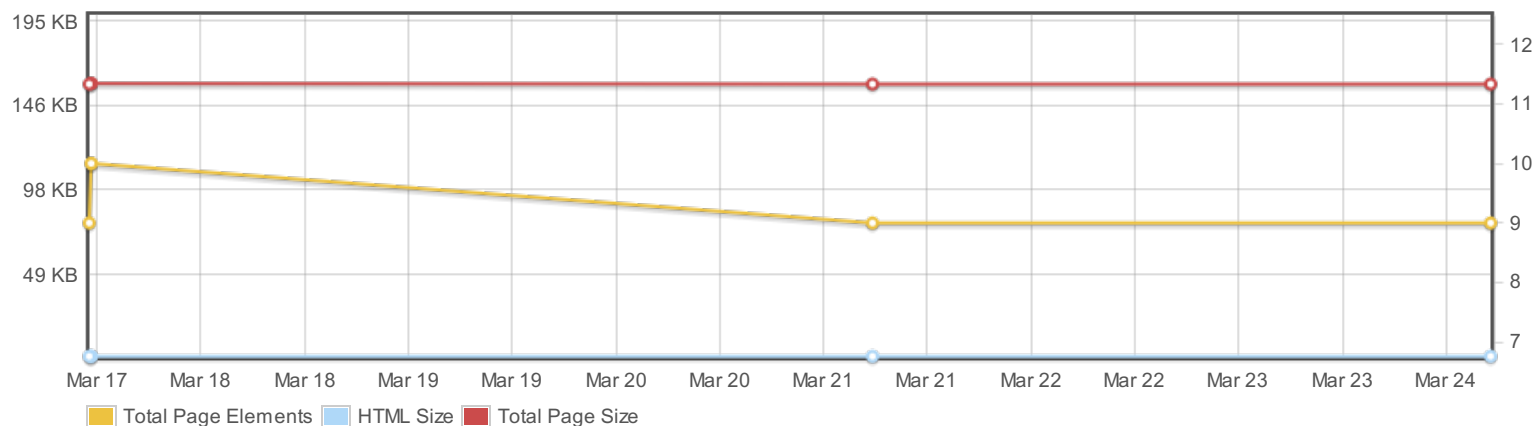


## History

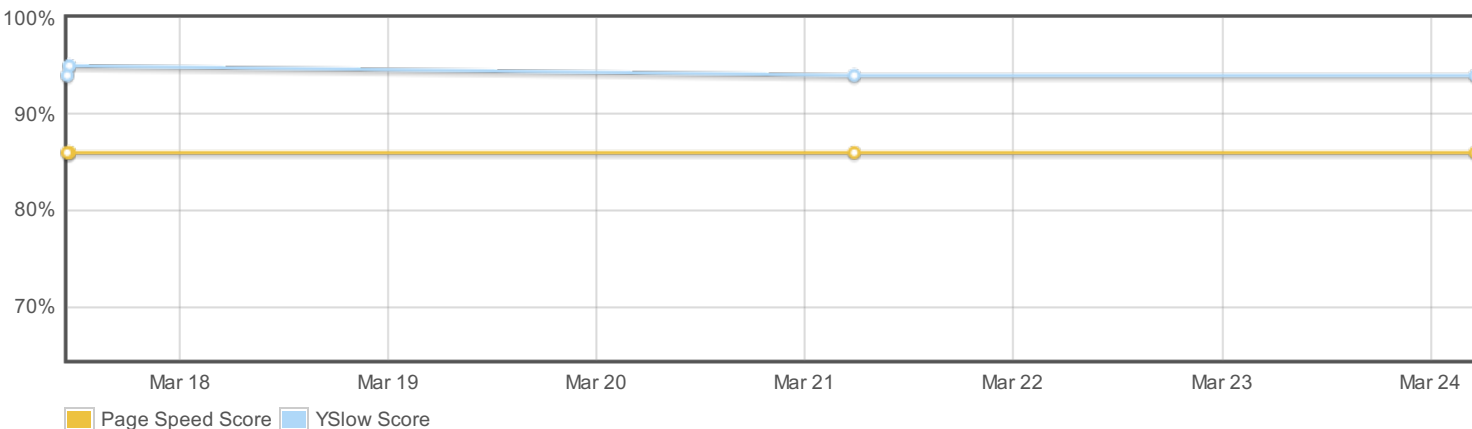
### Page load times



### Page sizes and request counts

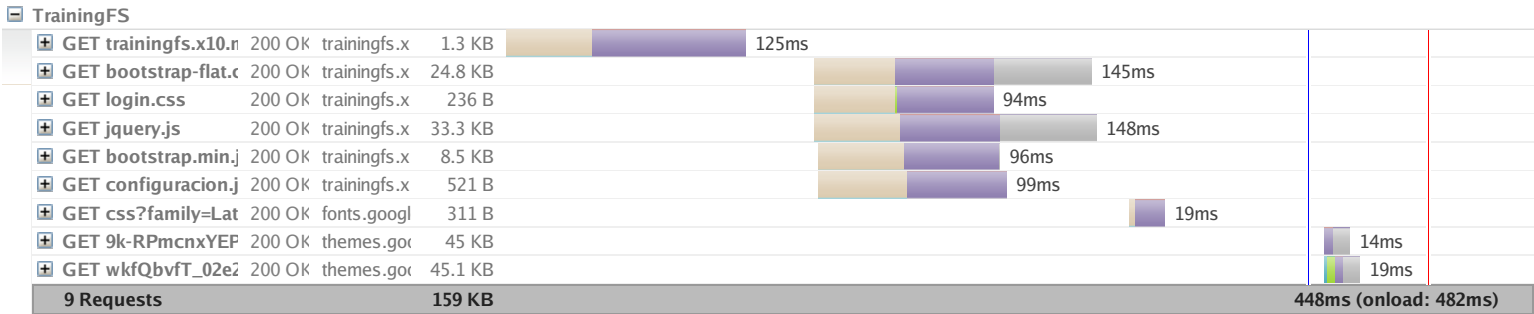


### Page Speed and YSlow scores



# Waterfall

The waterfall graph displays the loading behaviour of your site in Firefox. It can be used to discover simple issues such as 404's or more complex issues such as external resources blocking page rendering.



## Page Speed Recommendations

RECOMMENDATION	GRADE	RELATIVE	TYPE	PRIORITY
Inline small CSS	 D (67)	↓ Avg Score: <b>93%</b>	CSS	High
Enable Keep-Alive	 C (77)	↓ Avg Score: <b>96%</b>	Server	High
Specify a cache validator	 B (83)	↓ Avg Score: <b>94%</b>	Server	High
Avoid a character set in the meta tag	 B (85)	↓ Avg Score: <b>95%</b>	Content	High
Avoid CSS @import	 B (85)	↓ Avg Score: <b>96%</b>	CSS	Medium
Defer parsing of JavaScript	 B (88)	↑ Avg Score: <b>66%</b>	JS	High
Minify HTML	 A (91)	⬆ Avg Score: <b>94%</b>	Content	High
Specify a character set early	 A (95)	⬆ Avg Score: <b>98%</b>	Content	High
Minify CSS	 A (96)	↑ Avg Score: <b>84%</b>	CSS	High
Minify JavaScript	 A (99)	↑ Avg Score: <b>91%</b>	JS	High
Avoid bad requests	 A (100)	⬆ Avg Score: <b>97%</b>	Content	High
Avoid landing page redirects	 A (100)	⬆ Avg Score: <b>97%</b>	Server	High
Enable gzip compression	 A (100)	↑ Avg Score: <b>83%</b>	Server	High
Inline small JavaScript	 A (100)	⬆ Avg Score: <b>96%</b>	JS	High
Leverage browser caching	 A (100)	↑ Avg Score: <b>58%</b>	Server	High
Minimize redirects	 A (100)	↑ Avg Score: <b>93%</b>	Content	High
Minimize request size	 A (100)	⬆ Avg Score: <b>99%</b>	Content	High
Optimize images	 A (100)	↑ Avg Score: <b>74%</b>	Images	High
Optimize the order of styles and scripts	 A (100)	↑ Avg Score: <b>94%</b>	CSS/JS	High
Put CSS in the document head	 A (100)	⬆ Avg Score: <b>100%</b>	CSS	High
Remove query strings from static resources	 A (100)	↑ Avg Score: <b>89%</b>	Content	High
Serve resources from a consistent URL	 A (100)	↑ Avg Score: <b>94%</b>	Content	High
Serve scaled images	 A (100)	↑ Avg Score: <b>85%</b>	Images	High
Specify a Vary: Accept-Encoding header	 A (100)	↑ Avg Score: <b>90%</b>	Server	High
Specify image dimensions	 A (100)	↑ Avg Score: <b>53%</b>	Images	High
Combine images using CSS sprites	 A (100)	↑ Avg Score: <b>73%</b>	Images	Medium
Prefer asynchronous resources	 A (100)	↑ Avg Score: <b>94%</b>	JS	Medium

## YSlow Recommendations

RECOMMENDATION	GRADE	RELATIVE	TYPE	PRIORITY
Use a Content Delivery Network (CDN)	 E (50)	↑ Avg Score: 12%	Server	Medium
Add Expires headers	 B (89)	↑ Avg Score: 25%	Server	High
Configure entity tags (ETags)	 F (45)	↓ Avg Score: 58%	Server	Low
Use cookie-free domains	 C (75)	↑ Avg Score: 47%	Cookie	Low
Make fewer HTTP requests	 A (96)	↑ Avg Score: 43%	Content	High
Avoid empty src or href	 A (100)	⚡ Avg Score: 100%	Content	High
Compress components with gzip	 A (100)	↑ Avg Score: 71%	Server	High
Minify JavaScript and CSS	 A (100)	↑ Avg Score: 75%	CSS/JS	Medium
Avoid URL redirects	 A (100)	↑ Avg Score: 88%	Content	Medium
Make AJAX cacheable	 A (100)	⚡ Avg Score: 100%	JS	Medium
Put CSS at the top	 A (100)	⚡ Avg Score: 100%	CSS	Medium
Remove duplicate JavaScript and CSS	 A (100)	⚡ Avg Score: 100%	CSS/JS	Medium
Put JavaScript at bottom	 A (100)	⚡ Avg Score: 100%	JS	Medium
Avoid AlphasImageLoader filter	 A (100)	⚡ Avg Score: 96%	CSS	Medium
Avoid HTTP 404 (Not Found) error	 A (100)	⚡ Avg Score: 100%	Content	Medium
Reduce the number of DOM elements	 A (100)	↑ Avg Score: 94%	Content	Low
Do not scale images in HTML	 A (100)	⚡ Avg Score: 100%	Images	Low
Use GET for AJAX requests	 A (100)	⚡ Avg Score: 100%	JS	Low
Avoid CSS expressions	 A (100)	⚡ Avg Score: 97%	CSS	Low
Reduce DNS lookups	 A (100)	↑ Avg Score: 73%	Content	Low
Reduce cookie size	 A (100)	⚡ Avg Score: 100%	Cookie	Low
Make favicon small and cacheable	 A (100)	⚡ Avg Score: 100%	Images	Low
Make JavaScript and CSS external	(n/a)		CSS/JS	Medium

## Anexo III - Informe de usuarios

En este anexo se recogen informes de la opinión de futuros usuarios del sistema, entrenadores de fútbol sala de distintas categorías en competiciones nacionales.

### **Sergio Barroso Cáceres**

En primer lugar, se adjunta el análisis realizado por Sergio Barroso Cáceres, actual entrenador del Cádiz CF - Virgili FS y seleccionador del equipo masculino de fútbol sala de la Universidad de Cádiz.

### **Luis Alberto Ruiz López**

Seguidamente, se muestra el informe de análisis realizado por Luis Alberto Ruiz López, seleccionador provincial Cadete y Benjamín de la Selección Gaditana de Fútbol Sala.

### **Andrés Sanchez Romero**

Finalmente, se incluye el informe de opinión de Andrés Sanchez Romero, actual entrenador del Guadalcaén Fútbol Sala Femenino y entrenador de la Selección Femenina Gaditana de Fútbol Sala.

# INFORME DE OPINION

## *TrainingFS*

**Sergio Barroso Cáceres, Técnico Deportivo Nacional Fútbol Sala**

### Utilidad

En primer lugar comentar que esta aplicación es de gran utilidad para el gremio de técnicos-monitores de la modalidad de fútbol sala. Permite dar mayor seriedad y credibilidad a la labor de dichos monitores y técnicos, en cuanto pueden disponer de sus sesiones con una gran presentación. Además sobre todo de tener organizadas las propias sesiones y tareas. Todo ello online. Ayudando bastante este detalle al poco tiempo que suelen tener este gremio, que de esta forma pueden en cualquier hueco preparar sus sesiones.

### Comparativa

Con respecto a una comparativa con otro software o aplicaciones del mismo género, considero que TrainingFS tiene una gran fortaleza con respecto a la competencia: la relación “calidad resultado-tiempo empleado”. La mayoría del software específico suele caer en el mismo problema: suelen presentar tanta variedad de detalles insignificantes para adornar el producto, que se olvidan de lo realmente importante y el objetivo inicial. Así mismo suelen estar más enfocados a la parcela de la gestión de grupos o equipos. TrainingFS es una herramienta cuya aparente sencillez resulta brillante. ¿Motivo?. La gran mayoría, por no decir todos, los técnicos necesitan un editor de sesiones y tareas completo, intuitivo y sencillo de utilizar. Los que tienen tiempo para entrar en la gestión del club o equipo suele ser una minoría.

### Dificultad de uso

Ya hemos comentado en el apartado anterior que su principal fortaleza es que permite un gran resultado con un tiempo empleado mínimo. Gracias a que resulta tremendamente intuitivo. Sigue una nomenclatura y simbología estándar, lo cual reduce el tiempo que se tarda en manejar cualquier aplicación o programa. Antes comentamos su brillantez propiciada en que no tiene un icono o función de menos ni de más. Evitando distracciones que no llevan a ningún lado y propician que se nos olvide para qué usamos la aplicación: editor de tareas y sesiones.

## Valoración

Sinceramente creo que la idea y desarrollo de TrianingFS es potencialmente muy interesante incluso para intentar dar un salto empresarialmente (dentro de la humildad del mercado). Bien directamente o bien integrándolo en cualquier software específico deportivo. Creo que con ello se resume mi valoración de TrainingFS. Conozco muchísimos amigos técnicos y entrenadores en todo el panorama nacional, que estoy seguro que acogerían con gran entusiasmo este editor de tareas y sesiones.

## Sugerencias

- Estudiar integrarlo dentro de algún software de gestión deportiva.
- Traslación a otras modalidades deportivas de sala.

# Análisis del funcionamiento y utilidad de TrainingFS

El mencionado programa llegó a mi poder el pasado día 3 de febrero de 2014 como aplicación para facilitar la creación de sesiones de trabajo dentro de la planificación deportiva de la temporada.

Desde el punto de vista del funcionamiento del programa para preparar entrenamientos de fútbol sala debemos decir que nos parece un acierto la dualidad existente a la hora de poder crear tanto tareas de juego como para elaborar estrategias. Resultando de gran utilidad la pizarra que incorpora el programa y nos permite manejarnos tanto para desarrollar trabajos a pista completa como a media pista. Este detalle puede pasar desapercibido para los no conocedores de la forma de trabajar en nuestro deporte, pero para un usuario habitual de estos sistemas de trabajo podemos decir que es un instrumento realmente atractivo y eficaz.

El sistema de elaboración de contenidos es realmente novedoso, ya que permite crear multitud de ejercicios que se pueden agrupar en diferentes objetivos. Así, nos permite generar, a modo de ejemplo, un sinnúmero de tareas para trabajar distintas disciplinas del juego como pueden ser las transiciones; finalizaciones; repliegues; despliegues; y luego seleccionar a la hora de diseñar el entrenamiento cual/cuales de esas tareas previamente diseñadas queremos realizar.

El programa nos genera un doble beneficio desde mi punto de vista. Ya que tal como hemos comentado antes nos sirve para un doble fin:

- En primer lugar nos permite elaborar una copiosa base de datos agrupando ejercicios por categorías. Permitiendo que sea el usuario el que desarrolle dichos ejercicios a través de la pizarra.
- Por otro lado ayuda al entrenador a diseñar los microciclos de entrenamiento de forma ordenada y concisa, permitiendo crear los distintos entrenamientos y anotarlos en el calendario con el objetivo de saber qué objetivos se han ido trabajando a lo largo de la temporada para medir la efectividad de los mismos en función de los ejercicios trabajados.

Para finalizar este análisis sobre el programa “TrainingFS” diré que sin lugar a dudas facilita la vida al entrenador, y qué como herramienta de elaboración de tareas y creación de entrenamientos cumple a la perfección con su cometido.

Luis Alberto Ruiz López.

Seleccionador provincial Cadete y Benjamín de la Selección Gaditana de Fútbol Sala.



# INFORME - *TrainingFS*

**Andrés Sánchez Romero**

**Técnico de fútbol sala de nivel-1 y entrenador de la selección absoluta gaditana femenina**

Antes de comenzar me gustaría aclarar que no soy una persona aficionada a la informática y que he utilizado la herramienta desde una tablet.

## **Utilidad**

En mi opinión la aplicación que has diseñado me parece genial. En principio como herramienta de trabajo es súper útil, ya que simplifica el diseño de las tareas a realizar y jerarquiza los objetivos en función de las necesidades eventuales que se generan en un colectivo (equipo).

Poder visionar de forma inmediata las tareas y la ejecución de las mismas, consigue que los objetivos se alcancen con mayor prontitud y de una forma más eficiente. Tenemos que tener en cuenta que toda actividad que genere una dificultad de aprendizaje y comprensión debe ir complementada con instrumentos que faciliten ese aprendizaje. Esta aplicación lo consigue de forma óptima.

## **Opinión personal**

Para mi forma de trabajar, es una herramienta fundamental por lo que la voy a utilizar de forma asidua. Me gustaría que algún día añadieras algunos iconos más y le configurases el movimiento.

He conocido otros programas similares pero más complejos, por lo que me decanto por esta sin ningún tipo de duda.

¡Muchas gracias por esta herramienta!

Andrés Sánchez Romero